

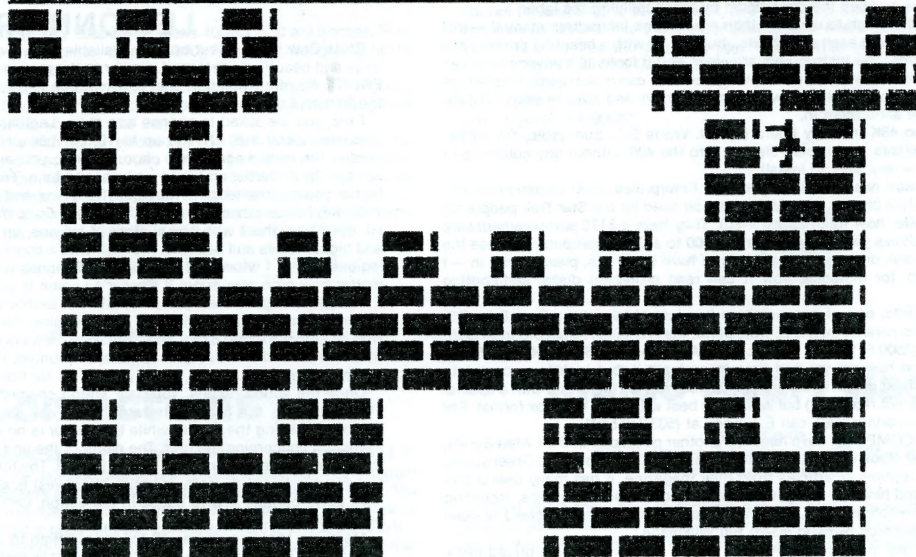
ACE  
ATARI  
COMPUTER  
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

-----  
FEBRUARY, 1983

Mike Dunn & Jim Bumpas, Editors  
-----

**KNIGHT FOR A DAY**





# News and Reviews

by Mike Dunn, Editor

I have just bought a copy of the brand new **Spell Wizard** by DataSoft (9421 Winnetka Ave., Chatsworth, CA 91311, \$80) and will review it completely next issue as I get to use it. This comes with a 33,000+ word dictionary, scans your text files at 600 words a minute and automatically identifies any misspelled words, allowing you to immediately correct them in-line. It is a stand alone utility which works without the word-processor, on any file made in DOS 2 format or converted to it such as Letter Perfect files with the Disk Utility. It is menu driven, and allows you to add any words you want, on the same file or on separate files. Looks like a good program, and for the price, a great bargain compared to similar programs for other computers. I ran this article through it and it seemed to work fine. It flagged the abbreviations and proper names, but allows you to add them to your dictionary. It even flagged Atari! Very impressive, more next issue.

**Letter Perfect** (by LJK, POB 10827 St. Louis, MO 63129 \$150 disk, \$200 ROM), has been quietly upgraded by adding significant new features without fanfare. L.J.K. is constantly upgrading all its programs, and unless you are really on your toes, it can pass by without comment. The latest version obtained by one of our club members is 3.1. You can now configure any printer on both the disk and ROM versions. Both 40 col. and 80 col. versions come with the disk version, and new commands have been added, such as an **Insert** mode which allows you to insert words directly rather than first make a space or line for it. L.J.K. has also just released a new documentation manual for **Data Perfect** which includes a 100 page tutorial and a price increase to \$129 after Feb. 15. In the last issue, Kirt Stockwell reviewed both Data Perfect and Filemanager 800+; he liked Data Perfect much better and felt it was more powerful and versatile, but because of the documentation and the general ease of use, concluded that Filemanager might be a better choice for a novice or for one whose needs are simple. This new documentation, which I have not seen, might very well make DataPerfect a better choice for a wider audience. Although L.J.K. can be congratulated for constantly striving to improve their products. They explain the upgrade policy as follows: Registered owners can obtain the Letter Perfect upgrade disk & documentation for \$30; Dataperfect upgrade documentation is \$10.

Incidentally, if any one has a program which will convert DataPerfect files to L.J.K. or vice versa, so we can exchange information with people with either database programs, please let me know.

For your Atari 400, several new products of interest. Two new keyboards are now available — never did receive one from Screen-Sonics. **Insoft software** (2485 Dunwin Dr., Mississauga, Ontario L5L 1T1, \$120 US) is a very easy to install full-stroke keyboard. You just remove the old membrane one and replace with the new one. The keys have a nice feel, but some are smaller than the 800 (eg. RETURN), and some are changed in location. I have not really used one, but some of our members have and liked it. **S/Ware** (1341 Quail Hollow Road, Harrisburg, PA 17112) has a series of outboard keyboards for the 400 and 800, including a deluxe one with a hexadecimal keypad for \$125. They use only high grade, commercial quality parts and are able to keep the price down by you supplying the labor.

They were kind enough to send me their nicely done instruction manual — this is really a kit — you wire each switch, etc., but end up with a beautiful product at a very attractive price. The instructions are clear, and it looks like anyone who can use a soldering iron fairly well and follow instructions can make themselves a nice keyboard. If any of you out there have any of the above and want to share your experiences, please send them in.

The new **Axlon** 48K memory board (170 N. Wolfe Rd., Sunnyvale, CA 94086, \$200) is now available. It can plug directly into the 400 without any soldering or wiring changes — very easy to install.

Another approach has been taken by **USS Enterprises** (6708 Landerwood Ln., San Jose, CA 95120) (If poor Jerry White can be sued by the Star-Trek people for **Triva Trek**, I wonder how this passes — ed). They have a \$175 software/hardware package which allows you to hook your 400/800 to a CP/M computer and use the CP/M keyboard, disk drives, etc. If any of you have done this, please write in — I don't understand, for instance, how it can read protected disks, self-booting disks, etc.

Speaking of CP/M, one of our members has just obtained the Disk Drive/RS-232C232/Centronics parallel Interface with CP/M, the ATR8000. Available from **Software Publishers** (2500 E. Randal Mill Rd. #125, Arlington, TX 76011, \$500 + \$250 for 64K and CP/M, he has not yet had time to evaluate it, but look for a complete evaluation in the next issue. It is supposed to read any CP/M format with a special program (which is not ready yet) but works the best with the 5 1/2 Xerox format. For those of you who cannot wait, call E.J. Knoll at (503)343-5191.

The Editors of **COMPUTE!** have released another new book in their Atari Series, **COMPUTES! First Book of Atari Graphics**, (Small Systems Services, Greensboro, NC \$13). Mostly reprints of articles from their magazine, it has many useful programs, utilities, and hints on using intermediate to advanced graphics, including P/M, character, animation, GTIA, etc. It also has one of the best articles I've seen explaining P/M graphics by (who else?) Bill Wilkinson.

Next month, I will do a comprehensive review on a moderately priced (\$80), ROM based word-processor from a big company. I have seen the documentation on it, and it looks like a winner!



# BUMPAS REVIEWS STRATOS

ADVENTURE INTERNATIONAL, Box 3435, Longwood, FL 32750 (no price given) produces **STRATOS**, a new arcade game. The game is a variation on the Missile Command theme: Waves of enemy attackers try to destroy your city.

The game has excellent color, and permits you to change background and foreground colors among a variety of choices by keyboard input. One of 2 sound selections may also be made.

The most significant feature of this game is in the animation of the attackers. They roll and spin, drop, dive and dodge your shots. With each change in attitude, the aspect of the ship changes on the screen. This gives them a '3D' effect.

Other features include a 'Repair Vehicle' which can be manipulated to repair broken spots in the 'force shield' which protects the city.

## SOUND & MUSIC

**Educational Software**, 4565 Cherryvale Ave., Soquel, CA 95073 produces **SOUND & MUSIC**, Tricky Tutorial #6 by our friend Jerry White. The price is \$19.95 and includes 'Player Piano'. The disk provides a very good step-by-step lesson program for using sound in BASIC routines. Player Piano will play back the selections you create. This is the same 'Player Piano' sold by APX for \$22.95. So you are getting a bargain here.

A 36-page digest sized booklet provides documentation, including program listings for several sample programs.

This work by Jerry White is presented with color & graphics, sound and text and may be valuable to anyone who wants to begin programming sound into their own software.

## SANDS OF EGYPT

(DataSoft, 9421 Winnetka Ave., Chatsworth, CA 91311. After getting the disk from Brian, three of us (Linda, Nick and myself) brainstormed the game on a Sunday afternoon. We've seen about a dozen screen pages with good color. The screen pages scroll smoothly, and don't just flash at you. Scrolling is used to simulate animation with cloud movement and a camel. So far, everything else is still. We've discovered about 60 of the 'more than 100 words' in the program's vocabulary. We've discovered about 16 of the objects in the game. I estimate we're only halfway through it, as we've just made it to the Pyramid. We still have to get in, find the treasure, and make it back to civilization.

The only way to effectively map the game is using the 'Hansel & Gretel' method suggested in the documentation. Movement otherwise is irrational and will lead to your death, lost in the desert. So far there seems to be only one correct path to any object, except for one or two 'shortcuts' (time & space warps?).

Some of the clues are very direct and will make the required action obvious. Some are very obscure. Most are just cute or seem irrelevant. I usually get impatient with adventure games because the vocabulary of the game seems to be the greatest obstacle one faces. This game seems to suffer less from this fault.

## LEGIONNAIRE

Chris Crawford's latest opus is available from Avalon Hill for \$35. The map is large and beautifully designed, similar to the map in his award-winning **EASTERN FRONT**. Players who know that game will be familiar with the joystick control of the Roman Legions in this game.

First, you are asked to choose how many Legions you want to use to fight the barbarians. You may choose up to 10, but beware! You must fight 2 barbarian hordes for each Legion you choose. Next, you are asked to choose the pair (always 2!) of barbarian tribes you want to face. There are 8 pairs of tribes. The higher you go, the tougher they get. The Huns and Helvetians are the toughest.

So far, I've only been able to defeat tribes up to the 4th-level Senones and Nervii. But I beat them with any number of legions, up to and including 10. The 5th and higher levels still beat me. I've won with a score of -8. I was down to my last 2 legions before I wiped out the last of 6 Senones and 6 Nervii. You can lose, no matter how high your score if Caesar's Legion is put out of action.

The game begins when you press START. Before you do, you may scroll across the map (I call it reconnaissance) to determine the location of the forces. The Roman Legions (up to 8 foot and 2 horse) are always grouped together in one location. The barbarian tribes (one is always mounted; the other is always foot) form two more separate groups, usually about as far from each other as from you.

Once the action begins, you hear sound seeming to simulate marching feet. The barbarians are coming towards you even as you give commands to the Legions. Pushing the trigger while the cursor is on a unit permits you to read information concerning the unit. The name of the unit will be the barbarian tribe, or the Roman leader in command of that Legion. The total number of men in the unit, and the number of 'swords' (combat effectiveness) is also shown.

SELECT permits you to pause in the action. OPTION permits you to quit and begin again.

Marching about causes a unit to lose strength to 'straggling'. These losses can be made up if you halt for a rest. Combat causes permanent losses. If you wait for the barbarians to attack you, you will find they halt a couple of impulses away. During this halt, they make a different noise. I think of barbarians beating their shields to raise themselves to a real gallic fury when I hear this sound. They are gathering their stragglers and preparing for the assault.

Each barbarian unit of one tribe starts with the same strength. The Legions vary in strength from 1800 (Labienus' cavalry) to 4000 (Caesar's Legion). To beat any but the 1st-level barbarians, you need to fight them when they are weak and you are strong. This is easier said than done. One way is to try to split their forces in such a way so you can concentrate the Legions on a smaller number of barbarian units before their friends can come to the rescue.

I plan to have many hours of fun with this game in the future.

—Jim Bumpas



# Benioff At Large

Greetings from the University of Southern California TKE Fraternity. This month, I have journeyed to the Las Vegas Winter Consumer Electronics Show (CES), received twelve new games from third party vendors, and was given three hours in a locked room with the new ATARI 1200XLS. So sit back, relax, and pour yourself a nice hot cup of coffee, and prepare yourself for another BENIOFF AT LARGE.

The night before my departure for Las Vegas, I realized my travel agent made a rather large mistake. Instead of booking me to Las Vegas and back to Los Angeles, she booked me going to Los Angeles, and back to Las Vegas. Unfortunately, all the airlines were booked, and I had to drive. I roared up the Supra, and took off for the Winter CES 1983.

As I arrived at the Las Vegas Convention Center, I realized I had a big job ahead of me. Not wasting any time, I headed straight for the ATARI BOOTH. New Products were being announced for the 400/800/1200XLS, 2600, and 5200 systems. All of these looked extremely inviting.

Besides the new 1200XLS (see below for detailed information), several pieces of hardware and software were announced for the home computers.

The all new ATARI printer is the Okidata Microliner 80, called the ATARI 1025 80 column printer. The retail price is \$549. It does not offer graphics, but will operate under Epson word processor programs. The new software from ATARI will support the new and old printers, and there should not be any problems (according to ATARI, not me). The casing matches the 1200xls (Black and White).

ATARI now has a color graphics 40 column color Plotter/Printer called the ATARI 1020. It retails for \$299. It is very sharp, and the graphics are great! You can also print up to 80 characters per line, with an option of 20 or 40 as well. Four colors can be used at once, with an option of eight colors from the set. It is a neat product, and the casing matches the 1200xls.

A new program recorder? YES! The all new ATARI 1010. It matches the 1200xls, and does everything the 410 does. It retails for \$99.95. I don't know the difference!

Several new software packages were released as well. They include several games and home management programs.

E.T. Phone Home! is the HomeE computer version of the VCS's E.T. It is very similar to the VCS, but better graphics, and more variations. Elliot! Help ATARI phone home.

Qix is an adaptation of the popular arcade game. Racing around filling up boxes is what it looks to be, but it is quite fascinating! Look for it in a couple months.

Dig Dug is excellent on the 400/800/1200 systems! It is a direct take from the arcade, and they did a very good job! It has clever graphics, and should be a big seller.

The New word processor from ATARI is the ATARI Writer. It is supposed to be excellent. It will work with all printers. It is menu driven, allows double columns, automatic page numbering, etc. It is very nice! \$79.95 on cartridge.

Timewise is a new calendar program. It allows you keep track of birthdays, appointments, etc. It looks very nice, and sells for \$29.95.

Also coming from ATARI is Family Finances, The Disney Educational Series, PAINT (a Micro Painter Rip-Off), Microsoft Basic II, Donkey Kong, and Superman III - The game.

New games for the 5200 (easily adaptable for the 400/800/120 are Soccer, Football, Baseball, Centipede (much better than 400/800), Kangaroo, Vanguard, Spabe Dungeon, and an awesome track ball.

New Games for the 2600 are Centipede, Ms. Pac-Man, Dig Dug, Vanguard, Galaxian, Phoenix, Real Sports Series, and a whole bunch of educational games from Sesame Street — Brian, you will like that!

A new ppo line joystick is also being released by ATARI. It looks very good, and will compete with Wico and Disc Washer. Also wireless joysticks will be available.

Mattel Electronics announced Intellivision II, a much sleeker unit, a new adapter for ATARI VCS games to play on Intellivision. What a deal!

Datasoft showed ATARI Zaxxon at the CES. They also had some other software released, but they were too busy to discuss it with me. See your dealer.

EMI released Orc Attack, River Rescue, Save the Seven Seas, Mutant Herd, and Major League Hockey all on cartridge.

Spectravideo announced a new computer and several new games for the ATARI 400/800. Vortex is a 3-D game which comes with special 3-D glasses. Also available are: Sector Alpha, Romper Room Series, Planet Patrol, China Syndrome, Nexar, Cave In, Number Crunch, REAGONOMICS!, Time Scape, Gold Mine, Ape Escape, Drive Em Krazy, Eagle Mountain, Master Cylinder, and Cosmic Math. This company is from Japan and looks excellent.

IMAGIC is releasing TWO games for the ATARI. Demon Attack, and Atlantis are now available in cartridge for the 400/800/1200.

This month I received twelve new games for the computer. I will try now to give brief descriptions of each game.

Kid Grid is very similar to Amidar in the arcades. It is from Tronix, a division of Softsel Distributing. I find it very fun, and challenging.

Speedway is a scrolling arcade game from IDS. I never knew cars could shoot at each other until I played this game. It is very interesting, but not extremely top notch in fascination.

Firebird is a new cartridge from Gebelli. I find it quite fun, but it is a bit simplistic, and gets boring at times.

Pogoman is a new game from Computer Magic. It is a very passive game. Just before I go to sleep at night I like this game because it is very relaxing. As you jump on your pogo stick, soft music plays in the back ground. It has super graphics!

Defender from ATARI is superb! If you like the arcade version, this is it! I played it over a year ago, and now you know why I liked it so much. It is truly awesome.

Sea Dragon is Russ Wetmore's new game from Adventure International. It is very similar to Scramble, if it were under water. I like this game, it even has the Preppie! character set.

Gorf is an excellent version of the arcade game from Roklan Corp. It is an exact duplication of the arcade version. Wizard of Wor is also by Roklan, and it is as exact as GORF.

London Software first released Space Ace, which was only fair. But, their new game, Hot Lips is excellent. As you chase around little men, you try to have them eaten by a huge mouth in the middle of the maze. I like this game.

Marauder is Sierra On-Line's new game. It is excellent!! It has two modes, and is like having two arcade games in one. An excellent Apple translation. This game is like advanced Missile Command.

Serpentine is Broderbunds new game for the ATARI. It is very compelling as you try to survive in a world with frogs and large snakes.

I have saved the best for last. Way Out from Sirius software blows my mind! It is like roller skating in a large maze. It has graphics better than Battlezone in the arcade, and has excellent game play — A super game for all. 3-D true to life graphics.

AND NOW THE 1200XLS. The next logical step? No. Sorry, the 1200 is just a 800 in a new case, and a few mord function keys. It is not even a true 64K! It is only available through machine language programming. It is very nice looking, if that counts for anything. It as 16K ROM, and 64K RAM. But, finally ATARI has a computer which LOOKS like a computer should look. It will retail for \$899, and will be in stores soon. You will definitely say, "Wow!" when you see it.

Good Bye From The City Of Smog.

—Marc Benioff  
631 W.28th Street  
Los Angeles, CA 90007  
(213)748-2458



## Corrections

The listing for **Scrambled Eggs** in the Dec newsletter won't work as it is printed, even if you get the control characters correctly entered. The program modifies the string P\$ pointers and requires it to be the first variable entered. The fix is to delete line 1 and re-enter it as line 11, then LIST the program out to tape or disk, type NEW and ENTER it back into the computer. RUN it and it will work.

The reason simply SAVeIng and LOADIng will not correct the problem is that these commands deal with tokenized programs; therefore the variable tables will not be reconstructed. An alternative fix is to re-enter the whole program, typing line 5 first, but I think the other method is much better.

—Dave Lutz, Holden, Alberta

*I wonder if this is why the program will not work if downloaded from the Bulletin Board in LIST form, but will if Tokenized and downloaded as a file.—ed.*

## Inside Atari Basic

by Bill Carris  
(\$12.95; Reston Publishing Co. 800-336-0338)  
Review by Larry Gold.

If you don't know how to program and are new to computers, this is a good book for you. It assumes you have no prior knowledge of computers or programming. If you are someone who knows computers and programming, but doesn't know the Atari, then it is a book for you. If you are someone who knows Atari Basic and knows how to use the computer, this will make a good reference book for color, GTIA, and the other graphic modes.

It is written in a simple style so all can understand, and makes no pretensions at being anything but what it is: A beginner's book. It is written to replace the Atari Basic book in use now. To go beyond the very basics, however, you will have to buy something more advanced.



# PROGRAMS

## WHAT IF?

(Conditionals In ATARI PILOT)  
by Ruth Ellsworth

One of the first problems we encountered when we started programming was writing logic statements. The IF - THEN conditional quickly became one of the most used and important concepts we learned. However, in ATARI PILOT the IF - THEN conditional appears, as my children say "in disguise."

The following is a list of IF - THEN statements which have been helpful to us.

TY: IF yes THEN type  
TN: IF no THEN type  
JM: IF there is a match THEN jump  
JY: IF yes THEN jump  
JN: IF no THEN jump  
CY: IF yes THEN compute  
CN: IF no THEN compute  
UY: IF yes THEN use  
UN: IF no THEN use  
EN: IF no THEN end

The IF - THEN conditional can also appear after a command word as follows:  
T(@B764=18):C IF (in this case, the last key typed has a decimal value of 18) THEN type C.

C(#X=1):\$Y=ZOO IF (the value of numeric variable X equals 1) THEN compute the value of \$Y equals ZOO.

J(#C×10):MODULE IF (#C is less than 10 THEN jump to \*MODULE.

U(#X=#Y):MODULE IF (#X equals #Y THEN use \*MODULE.

E(#X↑7): IF (#X is more than 7) THEN end.

The following programs illustrate the use of the conditionals to make easy games. The PRESCHOOL GAME chooses a letter at random which drops down the screen. The child must type a letter or return for the program to continue. The object is to type the matching letter from the keyboard before the letter drops "in to the water."

The LETTER SNATCH GAME is an improved and refined version of the PRESCHOOL GAME. It allows the child to choose a level of difficulty from 1 to 6, 1 being the easiest level. The letter which appears is in lower case, and the child must match the appropriate upper case letter from the keyboard. It is useful for matching lower and upper case letters and for practicing typing skills.



## COURTING CRICKETS

Clyde cricket lives in a factory. Deeply enamored of Cynthia cricket, he risks his life to bring her gifts, hoping to eventually win her feeler in marriage. Clyde must jump from conveyor to conveyor, avoiding any objects on them. He must also avoid any of the superstructure. To make Clyde jump you press stick 0 in the proper direction AND press the fire button, (both together).

To win Cynthia, Clyde must bring her flowers, perfume, candy, a necklace and finally a ring. He must return to the center of the lowest level to pick up the next gift. If Clyde fails he has three brothers willing to take his place. These brothers don't have to start at the beginning, they simply take over the gift delivery. Oh yes ... a former jealous suitor makes things difficult by throwing various objects from the top of the screen.

There are seven levels of difficulty. If you don't want that many, change line 450. The program as listed will not run in 16K. Leaving out the remarks will help, but other tricks will be necessary to scrunch it down.

—Stan Ockers



## KNIGHT FOR A DAY

(The instructions for this game are in the program listing.) **Program Description:**

Lines 10-24 are used to display the instructions.

Lines 90-190 reserve RAM for the new character set and transfer a combination of the old character set and data for the new characters. The data being from Lines 1000-1900

Lines 191-199 set up the original screen and reset all the variables to their original values.

Lines 200-699 contain the different sub-programs. Each one is a main program loop which is chosen depending upon the room of the castle into which you enter.

Lines 900-999 contain the separate subroutines to draw the different rooms and scenes contained in the game.

Lines 2000-2099 chose the random information to control which doorway leads to which room.

—Sydney Brown

# PM Graphics from BASIC

—Matt Giwer Annandale, VA

As everyone who has tackled the problem has found out, P-M Graphics is rather unpleasant to use. There are a dozen published approaches to the problem. However, they all involve machine language in some form or another.

Most of us are not machine language programmers and it takes a great motivation to get us to go to the effort. Also machine language routines are rarely general enough to provide the variety needed for different applications. And when the ones available are not what is needed it is back to the Assembler. Higher Order Languages, such as BASIC, are inherently more flexible but only really fast when they are compiled. Since BASIC instructions tend to be too flexible, so compilers can not interpret the meanings correctly, we are stuck with an interpreted language. This is slow, which means it can not be used for PM Graphics. Right? The only machine language routine in the enclosed article is to move the character down to where I want it in RAM. The DATA statements only put a message on the screen.

If you run this program you will see the words of the message printed vertically on the screen yet there are no print statements involved. One of the words scroll itself. Another word will do a circle (sort of) on the screen. The words will change very quickly. Note: there are some delays built into the program. You will see a surprise at the end.

The secret of this technique is the 30000 subroutine. For 1-line PMG a string of 4096 length is Dimensioned and cleared. (Development hint: instead of CHR\$(0) use CHR\$(255)) This program uses 5020 for the size since I wanted room for moving the character set. I am tired of creating new shapes. Since Atari BASIC reserves the length of the strings within the program space the manipulation of them is a simple machine language routine within the BASIC interpreter. There is very little housekeeping involved with this sort of manipulation as opposed to Microsoft BASIC which must change string locations when one gets longer.

Since screen output is handled by Direct Memory Access by ANTIC both the normal and the PM graphics can read from any location within the RAM. Normal graphics is slowed by the format requirements in location 87 and other instructions in the Display List. It can also read from any location. In exchange for the slow read, we get flexibility. When ANTIC reads the PM Graphics RAM specified by the number in PMBASE, 54279, it reads straight through without format or interpretation. In return for this simplicity we get speed.

However, since PMBASE can be specified by the user I specify it to be within the PMS that I dimension and clear. A short algorithm finds the lowest 2K page boundary within that string and sets PMBASE. Also it determines the offset between ADR(PMS) and the start of the Missile page. So far so good. It works.

All I do is position PMBASE within a cleared string. Next I manipulate substrings within this large string. And I get machine language speed movement of the shapes. The speed is still dependent upon BASIC interpreting the instruction in the first place but once interpreted the execution time is trivial. Ever notice how fast Atari BASIC handles strings? Here this speed makes shapes move through the PM graphics area. The speed of this technique is at least as fast as any machine language subroutine I have seen and it is in every way superior except for working out the arithmetic of the string manipulation. It is a simple bookkeeping problem.

What are the benefits? Page 6 is not used. (My kingdom for an Atari with ten Page sixes.) This technique works without change or modification in any Graphics mode. In other words you do not have to move RAMTOP or back off 8, 16, or 32 (!) pages to get PM Graphics to function properly. You forget about the problem entirely, it is not a problem here. In fact, since PM Graphics only uses five pages in single line, you can use the first three pages for other purposes as I have done here to define substrings. You can use the other parts of DIMensioned strings which are used for nothing else as I have used to hold the character set. You can use the bottom part below PMBASE if there is any. This method uses no machine language routines. Any thing you can define or manipulate from a string you can put on the screen.

Since this technique is independent of graphics mode you can write a display once and use it in many programs. How about a routine for dice? This can be used for anything from Monopoly and Craps to D&D.

Did I say dice? That takes at least six different shapes, twelve if you make red dice and white spots. So you define these shapes in other parts of the string and write them into the locations you want. You can have as many shapes as you have RAM to put them in. I moved down the character set to use so I have 128 different shapes for free. Also I add two more of my own design at the end of the program. So I have 130 shapes and full single line PM Graphics capability at a cost of a single DIMensioned string of 5020 characters in length. And if I look carefully for unused areas in the PMS I can find eleven pages free and still unused. (20 DIMensioned, -5 for PMG, -4 for Characters.)

The only hitch to all of this is ST must be determined while the program is running. The positions of the strings are only fixed when running. When in the direct mode the string locations move around. Thus you can only fine tune the PM display when it is running. Also all locations must be referenced to a value determined when the program is running. In this program the value is ST.

The scrolling part of the demo program shows this is better than called machine language routines. In this program I just scroll eight characters. However, with unnoticable time difference you can scroll the entire 256 bytes of the page with exactly the same technique used here. (Note there is a bug which works well at exactly 256 bytes. Use 255 or 257.)

This is an unbelievably simple technique yet I can find no hint of it in any of the literature. The Atari Hardware manual has thrown most people off in its instructions for turning on PMG. (You can make them work by making a GRAPHICS call AFTER you move RAMTOP.) From that point cut and try has led to this ungody method of moving the PMG area down below the screen RAM. The amount of RAM wasted is unbelievable and totally unnecessary. This method only works with Atari BASIC or with some language which keeps strings at the same location when a program is running. It probably will not work with Microsoft BASIC although I have not tried it. I have tried it quickly in all Graphics modes and in detail in modes 0, 1, 2, and 7 without problem. It does not work in 10 at all. (That is unsat! Any ideas?) But it does work in 9 and 11. Imagine your Players moving around in some of the GTIA landscapes.



```

100 REM #####
103 REM ###          ###
105 REM ### PM Graphics made Easy
107 REM ###          ###
110 REM ###          by          ###
113 REM ###          ###
115 REM ###          Matt Giwer  ###
117 REM ###          ###
120 REM #####
130 REM
150 REM ALL RIGHTS RESERVED
170 REM
1900 GOSUB 21610
1910 GOSUB 30000
1920 GOSUB 8000
2100 FOR IJK=1 TO 2 STEP 0
2205 PAGE=0:R=0
2210 READ A:IF A=999 THEN PAGE=PAGE+25
2215 R=R+1
2220 IF A=-1 THEN STEP 0
2230 PM$(PAGE+40+R*8,ST+PAGE+47+R*8)
      =PM$(ST+1280+A*8,ST+1287+A*8)
2240 GOTO 2210
2500 FOR I=0 TO 71
2510 PM$(ST-256,ST-253)=PM$(ST+256+48,
      ST+256+51)
2512 PM$(ST-200,ST-133)=PM$(ST+256+52,
      ST+256+119):PM$(ST-132,ST-129)=PM$(ST-
      256,ST-253)
2514 PM$(ST+256+48,ST+256+119)=PM$(ST-
      200,ST-129)
2516 IF I=9 OR I=45 THEN POKE 53259,1
2517 IF I=18 OR I=54 THEN POKE 53259,3
2518 IF I=27 OR I=63 THEN POKE 53259,1
2519 IF I=36 OR I=71 THEN POKE 53259,0
2520 NEXT I
2522 POKE 53259,0
2550 PM$(ST-256,ST-199)=PM$(ST+768+40,
      ST+768+97)
2555 DEC :FOR I=0 TO 90 STEP 2
2560 PM$(ST+768+40+I,ST+768+97+I)=PM$(
      ST-256,ST-199):POKE 53250,140+60*
      SIN(2*PI*I):NEXT I
2570 FOR I=91 TO 180 STEP 2
2575 PM$(ST+768+40+180-I,ST+768+97+180
      -I)=PM$(ST-256,ST-199):POKE 53250,140+
      60*SIN(2*PI*I):NEXT I
2580 PM$(ST-768,ST-681)=PM$(ST+48,ST+1
      36)
2581 PM$(ST-680,ST-593)=PM$(ST+256+48,
      ST+256+136)
2582 PM$(ST-592,ST-505)=PM$(ST+512+48,
      ST+512+136)
2583 PM$(ST-504,ST-417)=PM$(ST+768+48,
      ST+768+136)
2584 PM$(ST-416,ST-329)=PM$(ST+1024+48,
      ST+1024+136)
2595 PM$(ST+256+48,ST+256+136)=PM$(ST-
      768,ST-681)
2596 PM$(ST+512+48,ST+512+136)=PM$(ST-
      768,ST-681)
2597 PM$(ST+768+48,ST+768+136)=PM$(ST-
      768,ST-681)
2598 PM$(ST+1024+48,ST+1024+136)=PM$(S
      T-768,ST-681)
2600 FOR I=0 TO 500:NEXT I
2605 PM$(ST+256+48,ST+256+136)=PM$(ST-
      680,ST-593)
2606 PM$(ST+512+48,ST+512+136)=PM$(ST-
      680,ST-593)
2607 PM$(ST+768+48,ST+768+136)=PM$(ST-
      680,ST-593)

```

```

2608 PM#(5T+1024+48,5T+1024+136)=PM#(5
T-680,5T-593)
2610 FOR I=0 TO 500:NEXT I
2616 PM#(5T+512+48,5T+512+136)=PM#(5T-
592,5T-505)
2617 PM#(5T+768+48,5T+768+136)=PM#(5T-
592,5T-505)
2618 PM#(5T+1024+48,5T+1024+136)=PM#(5
T-592,5T-505)
2620 FOR I=0 TO 500:NEXT I
2627 PM#(5T+768+48,5T+768+136)=PM#(5T-
504,5T-417)
2628 PM#(5T+1024+48,5T+1024+136)=PM#(5
T-504,5T-417)
2630 FOR I=0 TO 500:NEXT I
2638 PM#(5T+1024+48,5T+1024+136)=PM#(5
T-416,5T-329)
2640 GOSUB 6000
2650 FOR I=0 TO 29
2655 T=PEEK(707):POKE 707,PEEK(706):PO
KE 706,PEEK(705):POKE 705,PEEK(704):PO
KE 704,PEEK(711)
2656 POKE 711,T:Q=I*1:NEXT I
2700 FOR I=18 TO 35:PM#(5T-750+I,5T-75
0+I)=CHR$(0):PM#(5T-700+I,5T-700+I)=CH
R$(0):NEXT I
2710 FOR I=0 TO 17:READ A:PM#(5T-750+I
,5T-750+I)=CHR$(A):NEXT I
2720 FOR I=0 TO 17:READ A:PM#(5T-700+I
,5T-700+I)=CHR$(A):NEXT I
2725 I=I+4
2730 PM#(5T+1250-I,5T+1285-I)=PM#(5T-7
50,5T-715)
2735 I=I+4
2740 PM#(5T+1250-I,5T+1285-I)=PM#(5T-7
00,5T-665)
2750 IF I>1280 THEN 2800
2760 GOTO 2725
2800 RESTORE
2900 NEXT I:K
6000 REM
6005 FOR I=0 TO 71
6010 PM#(5T-256,5T-253)=PM#(5T+512+48,
5T+512+51)
6012 PM#(5T-200,5T-133)=PM#(5T+512+52,
5T+512+119):PM#(5T-132,5T-129)=PM#(5T-
256,5T-253)
6014 PM#(5T+512+48,5T+512+119)=PM#(5T-
200,5T-129)
6020 NEXT I
6090 RETURN
8000 REM START MESSAGE
8010 POSITION 12,15:?"*, THIS IS A D
END"
8090 RETURN
9990 REM THE FOLLOWING DATA ARE THE CH
ARACTER NUMBERS IN OS CHBASE ORDER.
ref. TABLE 9.6 BASIC MANUAL
10000 DATA 45,97,116,116,0,39,105,119,
101,114,999
10010 DATA 112,114,101,115,101,110,116
,115,999
10020 DATA 48,45,0,39,50,33,48,40,41,3
5,51,999
10030 DATA 109,97,100,101,999
10040 DATA 54,37,50,57,0,37,33,51,57,9
99,-1
10050 DATA 0,66,66,102,102,239,251,251
,255,255,255,126,126,124,60,60,24
10055 DATA 24,60,126,126,255,251,25
1,255,255,255,255,126,126,124,60,60,24
21610 DIM M$(32)
21660 M$=""the following machine langua
arg string"":REM FOUR PAGE MOVE
21661 REM M$=hh [inv.contr.EJ]inv.UJ[
i.EJ]i.Jh[i.c.EJ]c.Wjh[i.c.EJ]i.V"J
[i.c.DJ]i.space[i.c.,]i.c.1J[i.c.Q]
21662 REM M$=continued [inverse all the
ese:VhPyfUfWJpJ]control .J

```

```

21990 RETURN
30000 REM PM SETUP
30005 DIM PM$(5020)
30010 PM$(1)=CHR$(0):PM$(5020)=CHR$(0)
      :PM$(2)=PM$(1)
30017 GRAPHICS 17
30020 POKE 559,62:POKE 623,49:POKE 532
      77,3
30030 POKE 53277,3
31000 REM FIND PMBASE
31010 ADHI=INT(ADR(PM$)/256):ADLO=ADR(
      PM$)-256*ADHI
31020 P=INT((ADHI+8)/8):REM 2-LINE (AD
      HI+4)/4
31022 POKE 54279,P*8:REM PMBASE//2-LIN
      E P*8
31030 ST=8*P*256+1024-256-ADR(PM$):REM
      OFFSET WITHIN PM$ FOR START OF MISCELL
      E PAGE//4*P*256+512-128-ADR(PM$) 2-LIN
      E
31035 FOR I=0 TO 3:POKE 53252+I,86-2*I
      :NEXT I
31040 FOR I=0 TO 3:POKE 53248+I,100+20
      *I:NEXT I
31045 FOR I=0 TO 3:POKE 704+I,24+16*I
      +9):NEXT I:POKE 710,0
31050 PCH=(P*8+8)*256:Z=USR(ADR(M*4),2
      24*256,PCH):REM ST+5*256 IS START OF C
      HARACTER SET DATA
31090 RETURN
32000 SAVE "D:BANNER"

```



# Ellsworth's PILOT

## PRESCHOOL GAMES by Ruth Ellsworth

```

1 R: PRESCHOOL
5 R: by Ruth Ellsworth
10 *START
20 C: $X=7\27
30 C($X=1): $X=A
40 C($X=2): $X=B
50 C($X=3): $X=C
60 C($X=4): $X=D
70 C($X=5): $X=E
80 C($X=6): $X=F
90 C($X=7): $X=G
100 C($X=8): $X=H
110 C($X=9): $X=I
115 C($X=10): $X=J
120 C($X=11): $X=K
130 C($X=12): $X=L
140 C($X=13): $X=M
150 C($X=14): $X=N
160 C($X=15): $X=O
170 C($X=16): $X=P
180 C($X=17): $X=Q
190 C($X=18): $X=R
200 C($X=19): $X=S
210 C($X=20): $X=T
220 C($X=21): $X=U
230 C($X=22): $X=V
240 C($X=23): $X=W
250 C($X=24): $X=X
260 C($X=25): $X=Y
270 C($X=26): $X=Z
280 J($X=0): *START
290 C:@B1373=16
300 C:@B1374=2
310 WRITE: S
320 POS: 4,9
325 R: all 's inverse
330 WRITE: S
340 POS: 8,1
350 WRITE: S, $X
360 READ: K, $Y
370 M: $X
380 JY: *RIGHT
390 WRITE: S, J
400 POS: 4,9
410 WRITE: S, .....
420 POS: 8,2
430 WRITE: S, $X
440 READ: K, $Y
450 M: $X
460 JY: *RIGHT
470 WRITE: S, J
480 POS: 4,9
490 WRITE: S, .....
500 POS: 8,3
510 WRITE: S, $X
520 READ: K, $Y
530 M: $X
540 JM: *RIGHT
550 WRITE: S, J
560 POS: 4,9
570 WRITE: S, .....
580 WRITE: S,
590 POS: 8,4
600 WRITE: S, $X
610 READ: K, $Y
620 M: $X
630 JM: *RIGHT
640 WRITE: S, J
650 POS: 4,9
660 WRITE: S, .....
670 POS: 8,5
680 WRITE: S, $X
690 READ: K, $Y
700 M: $X

```

```

710 JM: *RIGHT
720 WRITE: S, J
730 POS: 4,9
740 WRITE: S, .....
750 WRITE: S,
760 POS: 8,6
770 WRITE: S, $X
780 READ: K, $Y
790 M: $X
800 JM: *RIGHT
810 WRITE: S, J
820 POS: 4,9
830 WRITE: S, .....
840 POS: 8,7
850 WRITE: S, $X
860 READ: K, $Y
870 M: $X
880 JM: *RIGHT
890 WRITE: S, J
900 POS: 4,9
910 WRITE: S, .....
920 WRITE: S,
930 POS: 8,8
940 WRITE: S, $X
950 READ: K, $Y
960 M: $X
970 JM: *RIGHT
980 WRITE: S, J
990 POS: 4,9
1000 WRITE: S, .....
1010 POS: 8,9
1020 WRITE: S, $X
1030 WRITE: S,
1040 POS: 4,9
1050 WRITE: S, .....
1060 WRITE: S, J
1070 POS: 4,9
1080 WRITE: S, .....
1090 POS: 4,7
(inverse '3 inv spaces, repeat)
1100 WRITE: S, '
1110 POS: 6,5
(inv. '3 inv spaces, etc)
1120 WRITE: S, '
1130 PA: 60
1140 UNWE: $
1150 J: *START
1160 E:
1170 *RIGHT
1180 WRITE: S, J
1190 WRITE: S, 122222221
1200 WRITE: S, 1 1
1210 WRITE: S, Q1 ' 10
1220 WRITE: S, 1 0 1
1230 WRITE: S, 1 \_/_/ 1
1240 WRITE: S, \_/_/
1250 WRITE: S,
1260 WRITE: S, good! (inverse)
1270 PA: 100
1280 UNWE: $
1290 J: *START
1300 E:

```

## LETTER SNATCH by Ruth Ellsworth

```

5 R: LETTER SNATCH
10 *OPTION
15 T: CHOOSE A LEVEL OF DIFFICULTY FROM
1 TO 6 (1 IS EASIEST, 6 IS HARDEST)
20 A: #0
30 M: 1,2,3,4,5,6
40 JM: *ONE, *TWO, *THREE, *FOUR, *FIVE, *SIX
X
50 E:
60 *ONE

```

```

70 C: $Y=90
80 J: *START
90 E:
100 *TWO
110 C: $Y=60
120 J: *START
130 E:
140 *THREE
150 C: $Y=30
160 J: *START
170 E:
180 *FOUR
190 C: $Y=20
200 J: *START
210 E:
220 *FIVE
230 C: $Y=10
240 J: *START
250 E:
260 *SIX
270 C: $Y=5
280 J: *START
290 E:
300 *START
310 UNWE: $
320 C: #C=1
330 C: @B764=49
340 T:
350 C: @B710=142+7
360 C: $X=7\27
370 J($X=0): *START
380 J($X=1): *A
390 J($X=2): *B
400 J($X=3): *C
410 J($X=4): *D
420 J($X=5): *E
430 J($X=6): *F
440 J($X=7): *G
450 J($X=8): *H
460 J($X=9): *I
470 J($X=10): *J
480 J($X=11): *K
490 J($X=12): *L
500 J($X=13): *M
510 J($X=14): *N
520 J($X=15): *O
530 J($X=16): *P
540 J($X=18): *R
550 J($X=19): *S
560 J($X=20): *T
570 J($X=21): *U
580 J($X=22): *V
590 J($X=23): *W
600 J($X=24): *X
610 J($X=25): *Y
620 J($X=26): *Z
630 E:
640 *WRONG
650 50:15
660 PA: 48
670 50:0
680 50:15
690 PA: 48
700 50:0
710 50:15
720 PA: 32
730 50:17
740 PA: 16
750 50:18
760 PA: 48
765 J: *START
770 E:
780 *RIGHT
790 T:
800 T:
820 C: @B710=64+0
840 T:
850 T:
860 T:
870 T:

```

```

880 T: GNF's inverse $X
890 T: G & F inverse
900 T: $X$X$X
910 T:
911 T:
912 T:
913 T:
920 T: !!!!! !!!!! !!!!! !!!!!
(G inverse)
930 T: ! ! ! ! ! ! ! ! ! !
!
940 T: ! ! ! ! ! ! ! ! ! !
!
950 T: ! ! ! ! ! ! ! ! ! !
!
960 T: ! ! ! ! ! ! ! ! ! !
!
970 T: ! ! ! ! ! ! ! ! ! !
!
980 T: !!!!! !!!!! !!!!! !!!!!
(inverse F)
1050 PA: 100
1060 T:
1070 J: *START
1080 E:
1090 *A
1100 C: $X=A
1110 POS: 15,23
1120 T: .....
1130 POS: 20, #C
1140 T: a
1150 J(@B764=63): *RIGHT
1160 PA: $Y
1170 T:
1180 C: #C=#C+2
1190 J(#C(19)): *A
1200 POS: 17,18
1210 T:
1220 POS: 15,20
1230 T:
1240 POS: 15,23
1250 T: .....
1260 J: *WRONG
1270 E:
1280 *B
1290 C: $X=B
1300 POS: 15,23
1310 T: .....
1320 POS: 20, #C
1330 T: b
1340 J(@B764=21): *RIGHT
1350 PA: $Y
1360 T:
1370 C: #C=#C+2
1380 J(#C(19)): *B
1390 POS: 17,18
1400 T:
1410 POS: 15,20
1420 T:
1430 POS: 15,23
1440 T: .....
1450 J: *WRONG
1460 E:
1470 *C
1480 C: $X=C
1490 POS: 15,23
1500 T: .....
1510 POS: 20, #C
1520 T: c
1530 J(@B764=18): *RIGHT
1540 PA: $Y
1550 T:
1560 C: #C=#C+2
1570 J(#C(19)): *C
1580 POS: 17,18
1590 T:
1600 POS: 15,20
1610 T:

```

```

$X$X$X$X$X$X
$X $X
$X * $X
$X * $X

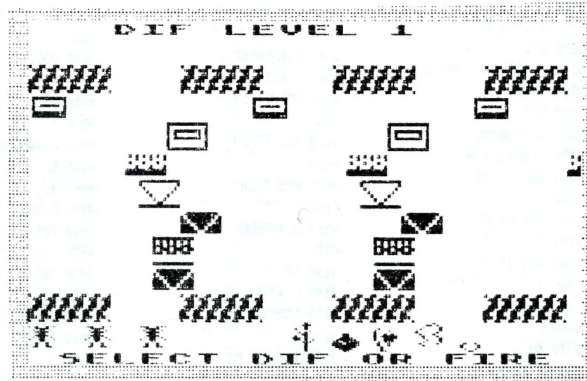
```







# Ockers: Fantastic new game!



## CRICKETS by Stan Ockers

```

1 REM *****
2 REM ** CRICKETS by Stan Ockers *
3 REM ** FEB 83 *
4 REM ** ACE NEWSLETTER *
5 REM ** 3662 VINE MAPLE *
6 REM ** EUGENE, OR 97405 **
7 REM ** $ 10 YR **
8 REM *****
100 REM *****
110 REM ** COURTING CRICKETS **
120 REM ** 5.0. 12-82 **
130 REM *****
140 ? "INITIALIZING ...."
150 DIM D$(1),F$(INT(ADR(D$)/2048)+1)
160 DIM BL$(13),CR$(12),CRJ$(12),CRF$(12),CRF2$(12),CRN$(12),STK$(32),HT1$(12),HT2$(12),SND$(173),HI=ADR(D$)/256
170 DIM GF1$(15),GF2$(15),DRP$(13),CUR$(12),COLDR$(4):GF1$="HIXLXHPQXTUXXXX":GF2$="JKXNOXR5XVXZCX"
180 REM ** Joystick Routine **
190 RESTORE 200:FOR J=1 TO 32:READ A:5
200 DATA 104,173,132,2,240,12,173,207,6,240,20,169,0,141,207,6,240,13,173,207,6,208,8,173,120,2,41,3,141,207,6,96
210 REM ** Sound Data **
220 RESTORE 230:FOR J=1 TO 173:READ A:
230 DATA 0.60,3,70,3,80,3,90,3,100,3,100,3,0,0,108,10,108,10,108,10,81,20,64,10,108,10,108,10,108,10,81,20
240 DATA 64,10,0,30,53,10,85,10,53,10,96,10,53,10,96,10,53,10,91,20,102,10,108,10,108,10,85,20,72,10,108,10
250 DATA 108,10,108,10,85,20,72,10,0,50,53,15,47,5,53,10,60,10,64,10,72,10,81,20,40,20,0,0
260 DATA 121,40,91,30,91,10,91,80,121,40,81,30,96,10,91,80,121,40,91,30,72,10,60,40,72,30,91,10,91,40,96,30
270 DATA 91,10,81,80,0,0,243,40,243,40,243,10,243,40,204,40,217,10,217,40
280 DATA 243,10,243,40,255,10,243,60,0,0,47,10,72,10,60,10,64,10,72,10,64,10,60,10,72,10,0,0
290 HI=INT(ADR(SND$)/256):POKE 209,HI:L5=ADR(SND$)-HI*256:POKE 208,L5:SOUND 3,0,0,0
300 DIF=1
310 REM ** DD$ is Screen Data **
320 DD$(1)="B":DD$(448)="B":DD$(12)=DD$(448):CHR$(0):DD$(828)=CHR$(0):DD$(450)=DD$(449)

```

```

330 LINE1=ADR(DD$)+513:HL=INT(LINE1/256):LL=LINE1-HL*256:POKE 88,LL:POKE 89,HL
340 POKE 559,0:GOSUB 1140:GOSUB 1330:GOSUB 1530
350 GRAPHICS 0:POKE 756,START/256:GOSUB 1380:POKE 559,0:POKE 560,0:POKE 561,6:POKE 559,34:GOSUB 1710
360 A=USR(1670):VERT=8:GOSUB 1780:POKE 88,LL:POKE 89,HL:POSITION 12,0:7 DIF:BR0=0
370 POKE 708,68:POKE 709,254:POKE 710,86:POKE 711,44:POKE 712,72:POSITION 16,7:7 "courting crickets"
380 POSITION 2,0:7 "dif level":RESTORE 382:FOR J=1 TO 4:READ A:COLDR$(J,J)=CHR$(A):NEXT J
382 DATA 228,36,4,230
390 DD$(1613,652)="YYYYYXXXXXXYYYYYXXXXXXYYYYYXXXXXXYYYYY"
400 DD$(1653,692)="YYYYYXXXXXXYYYYYXXXXXXYYYYYXXXXXXYYYYY"
410 DD$(693)="_XX_XX_":DD$(769)="abXXabXXabX":REM "=CONTROL"
420 GN=0:WFLG=0:GIFT=0:DROP=0:DD$(552)="XXXXXXXXXXXXXX":DD$(592)="XXXXXXXXXXXXXX"
430 POKE 1622,15:DD$(712)=GF1$:DD$(788)=GF2$
440 POSITION 16,7:7 "SELECT dif or FIRE":POKE 77,0
450 IF STRIG(0)=0 THEN 450
460 KEY=PEEK(53279):IF STRIG(0)=0 THEN 500
470 IF KEY<>5 THEN 460
480 DIF=DIF+1:IF DIF>7 THEN DIF=1
490 POSITION 12,0:7 DIF:GOSUB 1780:GOTO 460
500 POSITION 16,7:7 "COURTING CRICKET"
510 A=USR(ADR(STK$))
520 IF GIFT=1 AND YPOS=187 AND P>100 AND P<134 THEN GOSUB 820:GOSUB 790
530 IF PEEK(1743)=0 THEN FLAG=0
540 IF FLAG=1 THEN 580
550 S=PEEK(1740):IF S=2 OR S=1 THEN FL AG=1:P$(YPOS)=CRJ$:POKE 53767,170:POKE 1591,1:FOR J=1 TO 20:NEXT J
560 IF S=2 AND VERT=0 THEN POKE 1767+VERT,0:P$(YPOS)=BL$:YPOS=YPOS-16:P$(YPOS)=CR$:VERT=VERT-1:POKE 1767+VERT,1
570 IF S=1 AND VERT=8 THEN POKE 1767+VERT,0:P$(YPOS)=BL$:YPOS=YPOS+16:P$(YPOS)=CR$:VERT=VERT+1:POKE 1767+VERT,1
580 P=PEEK(1791):IF P>190 OR P<60 THEN POKE 1767+VERT,0:GOTO 710
590 POKE 53278,0

```

```

600 FOR J=1 TO 10:NEXT J
610 IF PEEK(53252)=0 THEN POKE 1767+VERT,0:GOTO 710
620 IF YPOS=59 AND P>100 AND P<134 THEN POKE 1767+VERT,0:GOTO 840
630 JPOS=YPOS+DELJ:IF JPOS=RTLJ OR JPOS<LLJ THEN DELJ=DELJ:JPOS=YPOS+2*DELJ
640 POKE 53250,JPOS
650 DRCNT=DRCNT-1:IF DRCNT<1 THEN DRCNT=10+5*(10-DIF):DPOS=50:GOSUB 950:P$(DPOS)=DRP$:DROP=1:POKE 53251,JPOS
660 IF DROP=1 THEN P$(DPOS)=BL$:DPOS=DPOS+DELTA:P$(DPOS)=DRP$:SOUND 1,DPOS-40,10,10
670 IF DPOS<240 THEN GOSUB 820:DPOS=50
680 IF PEEK(53260)=8 THEN POKE 1767+VERT,0:GOTO 710
690 GOTO 510
700 REM ** Falling Cricket **
710 GOSUB 820
720 P$(YPOS)=BL$:YPOS=YPOS+6:P$(YPOS)=CRF1$:SOUND 0,YPOS,10,10:FOR J=1 TO 30:NEXT J
730 P$(YPOS)=BL$:YPOS=YPOS+6:P$(YPOS)=CRF2$:SOUND 0,YPOS,10,10:FOR J=1 TO 30:NEXT J:IF YPOS<240 THEN 720
740 SOUND 0,0,0,0
750 BR0=BR0+1:IF BR0=4 THEN 1030
760 J=4*(BR0-1):DD$(693+J,696+J)="XXXX":DD$(769+J,772+J)="_X_X_"
770 VERT=8:GOSUB 1710:POKE 1791,120:POKE 1622,15:GOTO 510
780 REM ** Erase Next Gift **
790 GIFT=0:GN=GN+1:GOSUB 1000:DD$(712+GN*3)="XX":DD$(788+GN*3)="XX"
800 RETURN
810 REM ** Eliminate Drop **
820 SOUND 1,0,0,0:P$(DPOS)=BL$:POKE 53251,0:DROP=0:RETURN
830 REM ** Reached Female **
840 GOSUB 820:POKE 1791,118
850 IF GIFT=0 THEN GOSUB 920
860 IF WFLG=1 THEN 1090
870 FOR K=1 TO 10:P$(YPOS-14)=HT1$
880 FOR J=15 TO 0 STEP -1:SOUND 0,20,10,J:NEXT J:P$(YPOS-14)=HT2$:FOR J=1 TO 15:NEXT J:NEXT K
890 IF STRIG(0)=1 THEN 890
900 P$(YPOS)=BL$:P$(YPOS-14)=BL$:S=1:GOTO 560
910 REM ** Print Gift **
920 GIFT=1:DD$(552)=GF1$(1,(GN+1)*3):DD$(592)=GF2$(1,(GN+1)*3):IF GN=4 THEN WFLG=1
930 RETURN
940 REM ** Pick a Weapon **
950 R=INT(RND(0)*4):RESTORE 960+10*R:FOR J=1 TO 13:READ A:DRP$(J,J)=CHR$(A):NEXT J:POKE 707,ASC(COLDR$(R+1))
952 RETURN
960 DATA 20,72,34,20,74,40,8,127,127,62,62,28,28
970 DATA 0,0,80,112,112,112,112,112,120,126,94,0,0
980 DATA 0,0,48,96,64,127,127,64,224,224,0,0,0
990 DATA 60,24,24,24,60,126,223,215,247,255,127,126,60
995 REM ** Pick Another Gift **
1000 POKE 1622,155:POKE 707,92:POKE 53251,124+12*GN:FOR J=1 TO 5:P$(205)=CUR$:FOR K=1 TO 30:NEXT K
1010 P$(205)=BL$:FOR K=1 TO 30:NEXT K:NEXT J:RETURN
1020 REM ** No More Brothers **
1030 POKE 1622,131:POKE 53277,0:FOR J=53261 TO 53264:POKE J,0:NEXT J:GRAPHICS 18:POSITION 4,3:7 #6:"All Brothers"

```



```

1040 POSITION 6,4:7 #6:"Are Gone"
1050 POSITION 3,7:7 #6:"PRESS start TO
":POSITION 5,8:7 #6:"try again"
1060 IF PEEK(53279)<>6 THEN 1060
1070 GOTO 350
1080 REM ** Marriage Takes Place **
1090 POKE 1622,93:FOR L=0 TO 6:FOR K=0
TO 3:POS=64*L+16*K:DD$(POS+1)="XXXXcd
XXXXcdXXXX":NEXT K:NEXT L
1100 FOR J=1 TO 1000:NEXT J
1110 POKE 53277,0:FOR J=53261 TO 53264
:POKE J,0:NEXT J:GRAPHICS 18:POSITION
3,3:7 #6:"And They Lived"
1120 POSITION 1,4:7 #6:"Happily Ever A
fter":GOTO 1050
1130 REM * Change character set *
1140 DIM ZZ$(32):RESTORE 1150:FOR I=1
TO 32:READ A:ZZ$(I)=CHR$(A):NEXT I
1150 DATA 104,104,133,204,104,133,203,
104,133,206,104,133,205,162,4,160,0
1160 DATA 177,203,145,205,136,208,249,
230,204,230,206,202,208,240,96
1170 POKE 106,PEEK(106)-5:START=(PEEK(
106)+1)*256
1180 A=USR(ADR(ZZ$),57344,START):RESTO
RE 1200:FOR I=START+512 TO START+807:R
EAD A:POKE I,A:NEXT I
1190 RETURN
1200 DATA 0,0,0,0,0,0,170,0,63,58,57
,58,63,63,170,0,252,172,108,172,252,25
2,170
1210 DATA 0,255,170,85,170,255,255,170
,0,51,33,18,33,255,85,170,0,191,239,25
1,254,255,255,170
1220 DATA 0,255,215,215,215,190,235,17
0,0,254,251,239,191,255,255,170,0,1,32
,184,32,12,3,16
1230 DATA 64,208,64,224,184,224,192,19
2,116,28,3,0,0,0,0,200,238,248,192,1
92,192,192,192
1240 DATA 0,0,0,0,2,2,3,3,0,0,0,128,
128,192,192,13,55,219,222,223,55,13,3
1250 DATA 112,220,247,247,247,220,112,
192,0,0,0,58,234,234,233,229,0,0,0,40,
234,170,154,86
1260 DATA 233,233,57,58,58,14,14,13,86
,90,90,104,104,160,160,128,0,3,12,48,1
92,192,48,12
1270 DATA 48,204,3,3,3,12,48,12,12,3,1
2,48,14,2,2,0,3,3,12,32,160,160,128
1280 DATA 0,0,0,0,0,0,0,233,181,173,
183,222,122,94,107,2,10,2,4,16,16,4,1,
128,160,128,16,4,4,16,64
1290 DATA 0,0,3,15,15,204,63,15,0,0,0,
207,204,255,255,195,0,0,240,48,48,48,2
40,240
1300 DATA 0,0,84,5,17,5,1,5,0,0,21,80,
68,80,64,80,17,65,1,4,4,20,0,0
1310 DATA 68,65,64,16,16,20,0,0,40,190
,179,176,176,44,11,2,80,244,52,52,52,2
08,64,0
1320 REM ** VBI Routine **
1330 DIM VBI$(75):RESTORE 1340:FOR J=1
TO 75:READ A:VBI$(J)=CHR$(A):NEXT J
:VBI=ADR(VBI$):RETURN
1340 DATA 216,162,0,160,0,222,240,6,16
,42,189,224,6,157,240,6,189,232,6,240,
10,24
1350 DATA 173,255,6,125,248,6,141,255,
6,24,185,16,6,125,216,6,153,16,6,221,2
08,6,208,6
1360 DATA 189,200,6,153,16,6,200,200,2
00,232,224,7,144,201,173,255,6,141,0,2
08,32,57,6,32,88,6,76,98,228
1370 REM ** Display List in Page 6 **
1380 RESTORE 1390:FOR J=1536 TO 1679:R
EAD A:POKE J,A:NEXT J

```

```

1390 DATA 112,112,112,70,0,0,68,20,0,6
8,60,0,69,100,0,69,0,0,69,80,0,69,128,
0,69,208,0,69,0,0,69,80,0
1400 DATA 69,128,0,69,140,0,68,180,0
1410 DATA 68,0,0,70,40,0,65,0,6
1420 DATA 0,0,0,0,0,0,172,55,6,240,23,
206,56,6,16,18,177,208,141,6,210,200
1430 DATA 177,208,240,5,141,56,6,200,1
52,141,55,6,96
1440 DATA 0,0,172,86,6,240,40,206,87,6
,48,17,169,13,205,87,6,144,3,173,87,6,
9,160,141,5,210,208,18,177,208
1450 DATA 141,4,210,200,177,208,240,5,
141,87,6,200,152,141,86,6,96,104,160,0
,162,0,169,7,76,92,228
1460 RESTORE 1470:FOR J=1541 TO 1583:5
TEP 3:READ A:POKE J,HI+A:NEXT J
1470 DATA 2,2,2,2,0,0,0,0,1,1,2,2,3,3
1480 RESTORE 1490:FOR J=1736 TO 1791:R
EAD A:POKE J,A:NEXT J
1490 DATA 0,80,128,208,0,80,128,0,16,6
4,144,192,16,64,144,0,1,255,1,255,1,25
5,1,0
1500 DATA 20,12,9,12,15,18,21,0,0,0,0,
0,0,0,0,0,20,12,9,12,15,18,21,0,252,4,
252,4,252,4,252,120
1510 HV=INT(VBI/256):POKE 1674,HV:POKE
1672,VBI-256*HV
1520 RETURN
1525 REM ** PM Images **
1530 RESTORE 1540:FOR J=1 TO 12:READ A
:CR$(J,J)=CHR$(A):NEXT J
1540 DATA 231,60,90,60,24,60,90,153,24
,36,36,102
1550 RESTORE 1560:FOR J=1 TO 12:READ A
:CRJ$(J,J)=CHR$(A):NEXT J
1560 DATA 66,36,60,90,60,153,126,24,24
,60,66,195
1570 BL$(1)=CHR$(0):BL$(13)=CHR$(0):BL
$(2)=BL$
1580 RESTORE 1590:FOR J=1 TO 12:READ A
:CRF1$(J,J)=CHR$(A):NEXT J
1590 DATA 195,36,60,90,60,25,62,88,156
,36,38,96
1600 RESTORE 1610:FOR J=1 TO 12:READ A
:CRF2$(J,J)=CHR$(A):NEXT J
1610 DATA 195,36,60,90,60,152,124,26,5
7,36,100,6
1620 RESTORE 1630:FOR J=1 TO 12:READ A
:CRM$(J,J)=CHR$(A):NEXT J
1630 DATA 66,165,60,90,36,24,126,153,6
0,126,36,102
1640 RESTORE 1650:FOR J=1 TO 12:READ A
:HT1$(J,J)=CHR$(A):NEXT J
1650 DATA 0,216,248,248,112,32,0,27,31
,31,14,4
1660 RESTORE 1670:FOR J=1 TO 12:READ A
:HT2$(J,J)=CHR$(A):NEXT J
1670 DATA 0,27,31,31,14,4,0,216,248,24
8,112,32
1680 RESTORE 1690:FOR J=1 TO 12:READ A
:CUR$(J,J)=CHR$(A):NEXT J
1690 DATA 68,238,254,254,254,254,5,1
24,124,56,56,16
1700 REM ** PM Init. **
1710 P0$(1)=CHR$(0):P0$(256)=CHR$(0):P
0$(2)=P0$:YPOS=187:P0$(YPOS)=CR$
1720 P1$(1)=CHR$(0):P1$(256)=CHR$(0):P
1$(2)=P1$:P2$(1)=CHR$(0):P2$(256)=CHR$
(0):P2$(2)=P2$
1730 P3$(1)=CHR$(0):P3$(256)=CHR$(0):P
3$(2)=P3$:P1$(60)=CRM$:P2$(44)=CR$:JPD
5=100:DELJ=3:RTLJ=200:LLJ=50:DELTA=6
1740 POKE 54279,HI:POKE 559,62:POKE 53
277,3:POKE 53248,120:POKE 704,116:POKE
53249,126:POKE 705,92

```

```

1750 POKE 53250,100:POKE 706,20:POKE 7
07,0:POKE 623,1
1760 RETURN
1770 REM ** Packages on Belts **
1780 DD$(1)="Q":DD$(44B)="Q":DD$(2)=DD
$
1790 FOR J=1760 TO 1766:SPEED=INT(0)*I
9-DIF)*4+(7-DIF):IF DIF>3 THEN SPEED=5
PEED*1.8
1792 POKE J,SPEED:NEXT J
1800 FOR L=0 TO 6:RESTORE 1850+10*L:RE
AD F$:FOR K=0 TO 3
1810 POS=64*L+16*K:DD$(POS+1)=F$
1820 IF DIF>3 THEN DD$(POS+9)=F$
1840 NEXT K:NEXT L:RETURN
1845 REM ** CHAR. IN 1860,1880 & 1900
ARE INVERSE **
1850 DATA ACB
1860 DATA ACB
1870 DATA DDO
1880 DATA EFG
1890 DATA EFG
1900 DATA DDO
1910 DATA EFG

```



# Ruth's Pilot, Con't

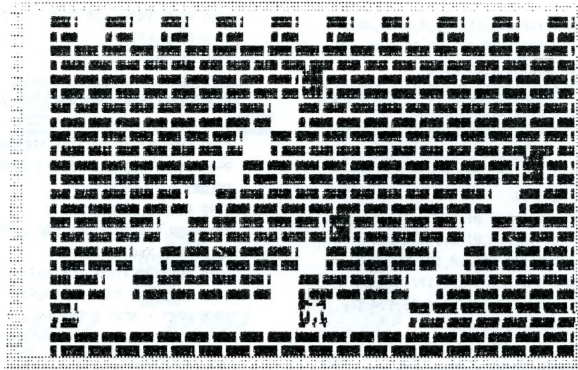
```

5540 T:
5550 POS:15,23
5560 T:
5570 J:WRONG
5580 E:
5590 *Y
5600 C:Y=Y
5610 POS:15,23
5620 T:
5621 POS:20,*C
5630 T:Y
5640 J(16764=43):*RIGHT
5650 PA:*Y
5651 T:
5660 C:Y=Y+2
5670 J(16719):*Y
5680 POS:17,18
5690 T:
5700 POS:15,20
5710 T:
5720 POS:15,23
5730 T:
5740 J:WRONG
5750 E:
5760 *Z
5770 C:Y=Y
5780 POS:15,23
5790 T:
5791 POS:20,*C
5800 T:Z
5810 J(16764=23):*RIGHT
5820 PA:*Y
5821 T:
5830 C:Y=Y+2
5840 J(16719):*Z
5850 POS:17,18
5860 T:
5870 POS:15,20
5880 T:
5890 POS:15,23
5900 T:
5910 J:WRONG
5920 E:

```



# Brown



## KNIGHTS by Sydney Brown

```

1 REM *****
2 REM ** KNIGHTS by Sydney Brown *
3 REM ** FEB 83 *
4 REM ** ACE NEWSLETTER *
5 REM ** 3662 VINE MAPLE *
6 REM ** EUGENE, OR 97405 **
7 REM ** $ 10 YR **
8 REM *****
10 ? "):POKE 710,194: ? ,:REM INV.: ? "
12 ? "KNIGHT FOR A DAY ": ? : ? "YOU ARE SIR
CYRIL THE BRAVE AND YOU"
14 ? "HAVE SET OUT TO RESCUE THE BEAUT
IFUL": ? "PRINCESS WHO WAS CAPTURED AND
LOCKED"
16 ? "IN THE TOWER OF THE EVIL KINGS C
ASTLE": ? : ? "YOU BEGIN IN THE ENTRANCE
CHAMBER OF": ? "THE CASTLE."
18 ? : ? "YOU MUST SEARCH ALL THROUGH T
HE VERY": ? "CONFUSING MAZE OF ROOMS A
ND FIND THE"
20 ? "THREE STAIRCASES, ONE LEADS TO AN
OTHER": ? "TOWER & YOU ARE SENT BACK TO
THE": ? "ENTRANCE CHAMBER TO BEGIN AGA
IN."
22 ? "ANOTHER LEADS TO THE DUNGEON WHI
CH": ? "ALSO LEADS BACK TO THE ENTRANCE
HALL."
24 ? "THE THIRD LEADS TO THE TRAPDOOR
ROOM,": ? "IF YOU MAKE IT PAST THIS ROO
M THEN YOU": ? "HAVE MADE IT."
26 ? "THE TIME YOU TAKE IS SHOWN AT
THE END": ? :REM INV.: ? "PRESS THE STA
RT BUTTON AND STAND-BY":
80 IF PEEK(53279)<6 THEN 80
90 POKE 106,PEEK(106)+2
95 GRAPHICS 18:POKE 708,30:POKE 709,14
0:POKE 710,188:A=PEEK(106)*256
100 FOR B=0 TO 511
101 IF (B/423) AND (B/456) OR B/463 THEN
READ D:POKE A+B,D:NEXT B:GOTO 110
106 POKE A+B,PEEK(57344+B):NEXT B
110 TH=0:POSITION 0,0: ? #6: ? "):POKE 71
0,10:POKE 712,0:POKE 708,44:POKE 709,2
22:POKE 711,140:POKE 756,PEEK(106)
190 GOSUB 2000
191 H=10:V=11:POSITION 0,11:REM INV.: ?
#6: ? "u":GOSUB 900:FOR W=0 TO 10:COLOR 2
45:PLOT W,11:FOR WW=1 TO 100:NEXT WW
192 COLOR 32:PLOT W,11:NEXT W:COLOR 24
5:PLOT 10,10:FOR W=1 TO 100:NEXT W:COL
OR 32:PLOT 10,10:COLOR 245:PLOT 10,9
193 FOR WW=1 TO 100:NEXT WW:COLOR 32:P
LOT 10,9:COLOR 245:PLOT 10,8:FOR WW=1
TO 100:NEXT WW:COLOR 32:PLOT 10,8
198 GOSUB 920:COLOR 245:PLOT H,V:X=B:Q
0=0

```

```

200 TH=TH+1:ST=STICK(0):IF ST=11 THEN
LOCATE H-1,V,Z:IF Z=32 THEN COLOR 32:P
LOT H,V:COLOR 245:H=H-1:PLOT H,V
202 POKE 77,0:IF ST=7 THEN LOCATE H+1,
V,Z:IF Z=32 THEN COLOR 32:PLOT H,V:COL
OR 245:H=H+1:PLOT H,V
204 IF ST=14 AND V=0 THEN LOCATE H,V-1
,Z:IF Z=32 THEN COLOR 32:PLOT H,V:COLO
R 245:V=V-1:PLOT H,V
206 IF ST=13 AND V=10 THEN LOCATE H,V+
1,Z:IF Z=32 THEN COLOR 32:PLOT H,V:COL
OR 245:V=V+1:PLOT H,V
270 IF V<1 THEN X=H:H=10:V=11:GOTO 290
279 FOR W=1 TO 77:NEXT W:GOTO 200
290 IF X(8) THEN XX=A:ON A GOSUB 925,93
0,935
291 IF X(11) THEN XX=B:ON B GOSUB 925,9
30,935
292 IF X(12) AND X(7) THEN XX=C:ON C GOS
UB 925,930,935
295 COLOR 245:PLOT H,V
300 QQ=0:ST=STICK(0):IF ST=11 AND H=0
THEN LOCATE H-1,V,Z:IF Z=32 THEN COLOR
32:PLOT H,V:COLOR 245:H=H-1:PLOT H,V
301 POKE 77,0
302 TH=TH+1:IF ST=7 AND H(19) THEN LOCA
TE H+1,V,Z:IF Z=32 THEN COLOR 32:PLOT
H,V:COLOR 245:H=H+1:PLOT H,V
304 IF ST=14 AND V=0 THEN LOCATE H,V-1
,Z:IF Z=32 THEN COLOR 32:PLOT H,V:COLO
R 245:V=V-1:PLOT H,V
306 IF ST=13 AND V=10 THEN LOCATE H,V+
1,Z:IF Z=32 THEN COLOR 32:PLOT H,V:COL
OR 245:V=V+1:PLOT H,V
340 IF H(1) OR H(18) OR V(1) THEN 355
350 FOR W=1 TO 77:NEXT W:GOTO 300
355 IF XX<>Q THEN 360
356 IF H(1) AND A=3 THEN GOSUB 940:GOTO
400
357 IF H(18) AND C=3 THEN GOSUB 940:GOT
0 400
358 IF V(1) AND B=3 THEN GOSUB 940:GOTO
400
360 IF H(1) AND XX=1 THEN ON A1 GOSUB 9
20,930,935
361 IF H(1) AND XX=2 THEN ON A2 GOSUB 9
25,920,935
362 IF H(1) AND XX=3 THEN ON A3 GOSUB 9
25,930,920
365 IF H(18) AND XX=1 THEN ON C1 GOSUB
920,930,935
366 IF H(18) AND XX=2 THEN ON C2 GOSUB
925,920,935
367 IF H(18) AND XX=3 THEN ON C3 GOSUB
925,930,920
370 IF V(1) AND XX=1 THEN ON B1 GOSUB 9
20,930,935
371 IF V(1) AND XX=2 THEN ON B2 GOSUB 9
25,920,935
372 IF V(1) AND XX=3 THEN ON B3 GOSUB 9
25,930,920

```

```

380 H=10:V=11:COLOR 245:PLOT H,V:IF QQ
=1 THEN QQ=0:GOTO 200
399 XX=X:GOTO 300
400 ST=STICK(0):TH=TH+1:POKE 77,0
401 IF ST=11 AND H(2) AND V=10 THEN COL
OR 32:PLOT H,V:COLOR 245:H=H+1:PLOT H,
V
402 IF ST=7 AND H(13) AND V=10 THEN COL
OR 32:PLOT H,V:COLOR 245:H=H+1:PLOT H,
V
404 IF ST=6 THEN LOCATE H+1,V-1,Z:IF Z
=122 OR Z=32 THEN COLOR 32:PLOT H,V:H=
H+1:V=V-1:COLOR 245:PLOT H,V
405 IF Z=122 THEN 450
406 IF ST=9 THEN LOCATE H-1,V+1,Z:IF Z
=32 THEN COLOR 32:PLOT H,V:COLOR 245:H
=H-1:V=V+1:PLOT H,V
449 FOR W=1 TO 77:NEXT W:GOTO 400
450 IF V=2 THEN ON A5 GOTO 480,490,550
452 IF V=5 THEN ON B5 GOTO 480,490,550
454 ON C5 GOTO 480,490,550
480 GOSUB 950:H=18:V=11:COLOR 245:PLOT
H,V:GOTO 600
490 SOUND 0,0,0:POKE 53768,5:POKE 53
761,168:POKE 53765,168:POKE 53760,254:
POKE 53764,127:GOSUB 960
495 COLOR 245:H=1:V=10:PLOT H,V
500 POKE 77,0:TH=TH+1:ST=STICK(0):IF S
T=6 AND V=3 THEN COLOR 32:PLOT H,V:H=H
+1:V=V-1:COLOR 245:PLOT H,V
502 IF ST=11 AND H(8) AND V=3 THEN COLO
R 32:PLOT H,V:H=H+1:COLOR 245:PLOT H,V
504 IF ST=7 AND H(19) AND V=3 THEN COLO
R 32:PLOT H,V:H=H+1:COLOR 245:PLOT H,V
530 IF H=19 THEN SOUND 0,0,0:0:0:SOUND 2
,0,0,0:GOTO 540
539 FOR W=1 TO 35:NEXT W:GOTO 500
540 POKE 712,0:GOSUB 920:H=10:V=11:COL
OR 245:PLOT H,V:GOTO 200
550 GOSUB 910:COLOR 245:PLOT 4,3:FOR W
=1 TO 7:R=AND(0)*220+5:SOUND 0,R,10,10
:SOUND 1,R+1,10,10:FOR WW=1 TO 77
589 NEXT WW:NEXT W:SOUND 1,0,0,0:SOUND
0,0,0,0:GOSUB 920:H=10:V=11:COLOR 245
:PLOT H,V:GOTO 200
600 TH=TH+1:ST=STICK(0):IF ST=11 THEN
LOCATE H-1,V,Z:IF Z=32 THEN COLOR 32:P
LOT H,V:COLOR 245:H=H-1:PLOT H,V
601 IF Z=123 THEN 490
602 IF ST=7 THEN LOCATE H+1,V,Z:IF Z=3
2 THEN COLOR 32:PLOT H,V:COLOR 245:H=H
+1:PLOT H,V
603 IF Z=123 THEN 490
604 IF ST=14 THEN LOCATE H,V-1,Z:IF Z=
32 THEN COLOR 32:PLOT H,V:COLOR 245:V=
V-1:PLOT H,V
605 IF Z=123 THEN 490
606 IF ST=13 AND V(10) THEN LOCATE H,V+
1,Z:IF Z=32 THEN COLOR 32:PLOT H,V:COL
OR 245:V=V+1:PLOT H,V
607 IF Z=123 THEN 490
610 IF V(1) THEN 650
649 FOR W=1 TO 35:NEXT W:GOTO 600
650 GOSUB 910:COLOR 32:PLOT 16,3:POSIT
ION 10,10: ? #6: ? "ux"
660 RESTORE 1900:FOR W=1 TO 14:READ F:
READ D:SOUND 0,F,10,10:SOUND 1,F+1,10,
10:FOR WW=1 TO D:NEXT WW:NEXT W:SOUND
0,0,0,0:SOUND 1,0,0,0
670 POSITION 6,0: ? #6: ? "time":TH
690 POSITION 4,11: ? #6: ? "press start"
695 IF PEEK(53279)<6 THEN 695
699 GOTO 110
900 GOSUB 910:POSITION 0,0: ? #6: ? "KNI
GHT FOR A DAY":SOUND 0,0,0,0:POKE 5376
8,4:POKE 53761,168:POKE 53765,168
902 POKE 53760,254:POKE 53764,127:FOR
WX=1 TO 7:FOR W=12 TO 252 STEP 16:POKE
710,W

```



```

905 FOR WM=1 TO 10:NEXT WM:NEXT W:NEXT
WX:POSITION 0,0:7 #6;"
"
909 SOUND 0,0,0,0:SOUND 2,0,0,0:RETURN

910 POSITION 0,0:7 #6;"":POKE 712,0:P
OKE 709,222:POKE 708,44:POKE 711,140:P
OKE 710,10
911 POSITION 2,2:7 #6;"***** ***
***:POSITION 3,3:7 #6;"* * * * * *x*
":POSITION 3,4:7 #6;"*****
912 POSITION 2,1:7 #6;"* * * * * *
"
913 POSITION 3,5:7 #6;"* * * * * *
":FOR W=6 TO 10:POSITION 3,W:7 #6;"*
*****:NEXT W
914 COLOR 32:PLOT 4,8:PLOT 6,8:PLOT 10
,8:PLOT 14,8:PLOT 16,8:FOR W=9 TO 11:P
LOT W,9:PLOT W,10:PLOT W,8:NEXT W
919 RETURN
920 POSITION 0,0:7 #6;"":Q0=1
921 POKE 708,44:GOSUB 990:COLOR 32:PL
T 4,0:PLOT 5,0:PLOT 9,0:PLOT 10,0:PLOT
14,0:PLOT 15,0:PLOT 9,11:PLOT 10,11
924 RETURN
925 POSITION 0,0:7 #6;"":X=1
926 POKE 708,90:GOSUB 990:COLOR 32:PL
T 0,5:PLOT 0,6:PLOT 9,0:PLOT 10,0:PLOT
19,5:PLOT 19,6:PLOT 9,11:PLOT 10,11
929 RETURN
930 POSITION 0,0:7 #6;"":X=2
931 POKE 708,202:GOSUB 990:COLOR 32:PL
OT 0,5:PLOT 0,6:PLOT 9,0:PLOT 10,0:PL
T 19,5:PLOT 19,6:PLOT 9,11:PLOT 10,11
934 RETURN
935 POSITION 0,0:7 #6;"":X=3
936 POKE 708,138:GOSUB 990:COLOR 32:PL
OT 0,5:PLOT 0,6:PLOT 9,0:PLOT 10,0:PL
T 19,5:PLOT 19,6:PLOT 9,11:PLOT 10,11
939 RETURN
940 POSITION 0,0:7 #6;"":POKE 708,44:
POKE 709,6:POSITION 0,0
941 ? #6;"*****:FOR W
=1 TO 11:7 #6;"*****:
NEXT W:COLOR 32:PLOT 10,8:DRAWTO 8,10
942 PLOT 9,3:DRAWTO 2,10:DRAWTO 13,10:
DRAWTO 17,6:COLOR 122:PLOT 10,2:PLOT 1
1,7:PLOT 18,5
949 H=10:V=10:COLOR 245:PLOT H,V:RETUR
N
950 POSITION 0,0:7 #6;"":POKE 708,14:
POKE 709,38:GOSUB 990:PLOT 6,1:DRAWTO
6,6:PLOT 13,10:DRAWTO 13,5
951 COLOR 32:PLOT 1,0:PLOT 2,0:PLOT 17
,11:PLOT 18,11
952 COLOR 123:PLOT 9,1:PLOT 10,10:PLOT
1,5:PLOT 18,6:FOR W=1 TO 14
953 Y=INT(RND(0)*8)+2:X=INT(RND(0)*16)
+2:LOCATE X,Y,Z:IF Z<32 THEN 953
954 PLOT X,Y:NEXT W
959 RETURN
960 POSITION 0,0:7 #6;"":POKE 712,4:P
OKE 710,0:POKE 708,10
961 W=8:COLOR 94:FOR Y=4 TO 10:FOR X=W
TO 19:PLOT X,Y:NEXT X:W=W+1:NEXT Y:CO
LOR 219:FOR W=4 TO 16 STEP 4:FOR X=W T
O W+2
962 PLOT X,9:PLOT X,10:NEXT X:NEXT W:P
LOT 11,6:PLOT 15,6:COLOR 218:PLOT 19,3
969 RETURN
990 COLOR 94:PLOT 0,0:DRAWTO 19,0:DRAW
TO 19,11:DRAWTO 0,11:DRAWTO 0,0:RETURN

999 RETURN
1000 DATA 28,8,62,93,93,20,20,54,1,3,7
,15,31,63,127,255,255,127,63,31,15,7,3
,1

```

```

1005 DATA 0,8,120,8,28,62,8,24,126,126
,126,126,94,126,126,126,170,255,170,17
0,170,170,255,170
1010 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0
1050 DATA 253,253,253,0,223,223,223,0,
0,0,0,0,0,0,0,0
1900 DATA 49,75,53,60,75,25,59,50,65,5
0,75,50,90,50,100,50,110,25,100,25,90,
50,100,25,90,25,80,150
2000 A=INT(RND(0)*3)+1
2001 B=INT(RND(0)*3)+1:IF B=A THEN 200
1
2002 C=INT(RND(0)*3)+1:IF C=A OR C=B T
HEN 2002
2010 A1=INT(RND(0)*3)+1
2011 B1=INT(RND(0)*3)+1:IF B1=A1 THEN
2011

```

```

2012 C1=INT(RND(0)*3)+1:IF C1=A1 OR C1
=B1 THEN 2012
2020 A2=INT(RND(0)*3)+1
2021 B2=INT(RND(0)*3)+1:IF B2=A2 THEN
2021
2022 C2=INT(RND(0)*3)+1:IF C2=A2 OR C2
=B2 THEN 2022
2030 A3=INT(RND(0)*3)+1
2031 B3=INT(RND(0)*3)+1:IF B3=A3 THEN
2031
2032 C3=INT(RND(0)*3)+1:IF C3=A3 OR C3
=B3 THEN 2032
2040 A5=INT(RND(0)*3)+1
2041 B5=INT(RND(0)*3)+1:IF B5=A5 THEN
2041
2042 C5=INT(RND(0)*3)+1:IF C5=A5 OR C5
=B5 THEN 2042
2050 Q=INT(RND(0)*3)+1
2099 RETURN

```

\*\*\*\*\*

### MACHINE LANGUAGE 3 by Stan Ockers

```

10 ; MOVING CURSOR ROUTINE
20 ;
30 ; The cursor position is held in page zero
40 ; locations $54 (row) and $55 (column)
50 ; Joystick 0 values appear at $0278
60 ;
0278 70 STICK0 = $278
0054 80 ROWCRS = $54
0055 90 COLCRS = $55
0100 ;
0000 0110 *= $0600
0120 ;
0600 68 0130 PLA ; for USR
0601 A07B02 0140 LDA STICK0 ; read Joystick
0604 C90E 0150 CMP #0E ; is it UP?
0606 D002 0160 BNE DN ; no
0608 C654 0170 DEC ROWCRS ; yes, cursor up one
060A C90D 0180 DN CMP #0D ; is it DOWN?
060C D002 0190 BNE LEFT ; no
060E E654 0200 INC ROWCRS ; yes
0610 C90B 0210 LEFT CMP #0B ; is it LEFT
0612 D002 0220 BNE RIGHT
0614 C655 0230 DEC COLCRS
0616 C907 0240 RIGHT CMP #07 ; is it RIGHT?
0618 D002 0250 BNE OUT
061A E655 0260 INC COLCRS
061C 60 0270 OUT RTS ; back to Basic
0280 ;
0290 ; Listing #1
0300 ;
0310 ;
0320 ; ANSWERS TO PROBLEMS
0330 ;
0340 ; Problem #3: Fill Screen with A's
0350 ;
061D 0360 *= $0600
0600 68 0370 PLA ; as usual
0601 A004 0380 LDY #04 ; four pages
0603 A921 0390 LDA #21 ; an ASCII 'A'
0605 A200 0400 LDX #0 ; start at zero ...
0607 CA 0410 LOOP1 DEX ; and go thru all ...
0608 9D407C 0420 STA $7C40,X ; (inserting A's) ...
060B D0FA 0430 BNE LOOP1 ; until zero again
060D EE0A06 0440 INC $060A ; HI byte up one
0610 88 0450 DEY ; ck for last page
0611 D0F4 0460 BNE LOOP1 ; not yet
0613 F0FE 0470 LOOP2 BEQ LOOP2 ; endless loop
0480 ;
0490 ; Notes: (1) My screen starts at
0500 ; $7C40 (32K), yours may differ.
0510 ; (2) We actually try to fill 1024 bytes
0520 ; (screen has 960) but screen data
0530 ; is at very top of memory. The extra

```



# Machine Language 3, Con't

```

0540 ; doesn't overrun anything.
0550 ; (3) Notice that you can't re-run
0560 ; the program without fixing it up
0570 ; where it modifies itself.
0580 ;
0590 ;
0600 ; Problem #4: Execution Time of Loops
0610 ;
0615 0620      *= $0600
0600 68      0630      PLA
0601 A914    0640      LDA #14      ; 20 (decimal) outermost loops
0603 85D0    0650      STA #D0      ; using page zero cntr.
0605 A200    0660 LOP3      LDX #0      ; init. loops
0607 A000    0670 LOP2      LDY #0
0609 88      0680 LOP1      DEY          ; innermost loop
060A D0F0    0690          BNE LOP1
060C CA      0700          DEX          ; second loop
060D D0F8    0710          BNE LOP2
060F C6D0    0720          DEC #D0      ; outermost loop
0611 D0F2    0730          BNE LOP3
0613 60      0740          RTS          ; back to Basic
0750 ;
0760 ; Note: Takes approx. 6 sec.
0770 ; for about 1,3 million inner loops
0780 ; That's half a million loops a sec. !

=027B STICK0      =0054 POWCR5      =0055 COLCR5      060A DN
0610 LEFT          0616 RIGHT      061C OUT          0607 LOOP1
0613 LOOP2          0605 LOP3      0607 LOP2          0609 LOP1

```

## Jon Attack

```

00100 ; FINE SCROLLING
00110 ;
00120 ; by Jon Attack
00130 ;
D300: 00140 STICK .EQ #D300
D405: 00150 VSCROL .EQ #D405
D404: 00160 HSCROL .EQ #D404
00170 .OR #690 STARTS AT PAGE SIX
00180 ;
0600: A9 03 00190 START LDA #3 TRY DIFFERENT NUMBERS HERE
0602: 8D E1 06 00200 STA SPEED
0605: AD 30 02 00210 LDA #230 SET UP LMS POINTER
0608: 8D CF 06 00220 STA LMS
060B: AD 31 02 00230 LDA #231 #230, #231 IS POINTER TO START
060E: 8D D0 06 00240 STA LMS+1 OF DISPLAY LIST
0611: A9 CB 00250 LDA #D5P CHANGE TO OUR
0613: 8D 30 02 00260 STA #230 CUSTOM DISPLAY LIST
0616: A9 06 00270 LDA /DSP
0618: 8D 31 02 00280 STA #231
061B: A9 06 00290 LDA #6 SET UP VERTICAL BLANK INTERRUPT
061D: A2 06 00300 LDX /VBI HI-BYTE
061F: A0 2D 00310 LDY /VBI LO-BYTE
0621: 20 5C E4 00320 JSR #E45C
0624: A9 00 00330 LDA #0
0626: 8D DF 06 00340 STA HSCRL
0629: 8D E0 06 00350 STA VSCRL
062C: 60 00360 RTS RETURN TO USER
00370 ;
00380 ; VERTICAL BLANK INTERRUPT
00390 ;
062D: AC E1 06 00400 VBI LDY SPEED
0630: AD 00 D3 00410 LOOP LDA STICK
0633: 29 02 00420 AND #2
0635: D0 20 00430 BNE NOTDOWN
0637: AE E0 06 00440 LDX VSCRL SCROLL DOWN
063A: E8 00450 INX
063B: E0 10 00460 CPX #16
063D: 90 12 00470 BCC .1
063F: AD CF 06 00480 LDA LMS
0642: 18 00490 CLC
0643: 69 18 00500 ADC #24
0645: 8D CF 06 00510 STA LMS
0648: C9 18 00520 CMP #24

```

```

064A: B0 03 00530 BCS .2
064C: EE D0 06 00540 INC LMS+1
064F: A2 00 00550 .2 LDX #0
0651: BE E0 06 00560 .1 STX VSCRL
0654: BE 05 D4 00570 STX VSCROL
0657: AD 00 D3 00580 NOTDOWN LDA STICK
065A: 29 01 00590 AND #1
065C: D0 1E 00600 BNE NOTUP
065E: AE E0 06 00610 LDX VSCRL SCROLL UP
0661: CA 00620 DEX
0662: 10 12 00630 BPL .1
0664: AD CF 06 00640 LDA LMS
0667: 38 00650 SEC
0668: E9 18 00660 SBC #24
066A: 8D CF 06 00670 STA LMS
066D: C9 E8 00680 CMP #232
066F: 90 03 00690 BCC .2
0671: CE D0 06 00700 DEC LMS+1
0674: A2 0F 00710 .2 LDX #15
0676: BE E0 06 00720 .1 STX VSCRL
0679: BE 05 D4 00730 STX VSCROL
067C: AD 00 D3 00740 NOTUP LDA STICK
067F: 29 04 00750 AND #4
0681: D0 1A 00760 BNE NOTLFT
0683: AE DF 06 00770 LDX HSCRL SCROLL LEFT
0686: CA 00780 DEX
0687: 10 0E 00790 BPL .1
0689: AE CF 06 00800 LDX LMS
068C: E8 00810 INX
068D: BE CF 06 00820 STX LMS
0690: D0 03 00830 BNE .2
0692: EE D0 06 00840 INC LMS+1
0695: A2 07 00850 .2 LDX #7
0697: BE DF 06 00860 .1 STX HSCRL
069A: BE 04 D4 00870 STX HSCROL
069D: AD 00 D3 00880 NOTLFT LDA STICK
06A0: 29 08 00890 AND #8
06A2: D0 1E 00900 BNE DONE
06A4: AE DF 06 00910 LDX HSCRL SCROLL RIGHT
06A7: E8 00920 INX
06A8: E0 08 00930 CPX #8
06AA: 90 10 00940 BCC .1
06AC: AE CF 06 00950 LDX LMS
06AF: CA 00960 DEX
06B0: BE CF 06 00970 STX LMS
06B3: C9 FF 00980 CMP #255
06B5: D0 03 00990 BNE .2
06B7: CE D0 06 01000 DEC LMS+1
06BA: A2 00 01010 .2 LDX #0
06BC: BE DF 06 01020 .1 STX HSCRL
06BF: BE 04 D4 01030 STX HSCROL
06C2: 88 01040 DONE DEY
06C3: F0 03 01050 BEQ .1
06C5: 4C 30 06 01060 JMP LOOP
06C8: 4C 5F E4 01070 .1 JMP #E45F EXIT VBI ROUTINE
01080 ;
01090 ; CUSTOM DISPLAY LIST
01100 ;
06CB: 70 70 70 01110 DSP .HS 707070 3 BLANK B'S
06CE: 77 01120 .HS 77
06CF: 20 BC 01130 LMS .HS 20BC LOAD MEMORY SCAN INSTRUCTION
06D1: 37 37 37
06D4: 37 37 37
06D7: 37 37 37
06DA: 37 17 01140 .HS 37373737373737373737
06DC: 41 01150 .HS 41
06DD: CB 06 01160 .DA DSP BACK TO START OF DISPLAY LIST
06DF: 00 01170 HSCRL .HS 00
06E0: 00 01180 VSCRL .HS 00
06E1: 00 01190 SPEED .HS 00

```

### --- Symbol table ---

06C2: DONE	.02=064F, .01=0651	06E1: SPEED
.01=06C8	0657: NOTDOWN	0600: START
06CB: DSP	.02=0674, .01=0676	D300: STICK
06DF: HSCRL	069D: NOTLFT	062D: VBI
D404: HSCROL	.02=068A, .01=06BC	06E0: VSCRL
06CF: LMS	067C: NOTUP	D405: VSCROL
0630: LOOP	.02=0695, .01=0697	



# Game Writer's Column

by Jon Attack, Eugene A.C.E.

Welcome back to the GAME WRITER'S COLUMN. Last month we covered Vertical Blank Interrupts. This month we'll be using VBI together with this month's topic: Vertical and horizontal fine scrolling.

To achieve fine scrolling, there are basically three things you must do:

1. Make a custom display list.
2. Set up a background to be scrolled.
3. Write a VBI routine to do the scrolling.

A Display List is a set of instructions telling the ANTIC chip (1) what part of memory is to be displayed and (2) how to display it. For our purposes we'll use a display list to scroll the whole screen both vertically and horizontally.

Hex:	Decimal:	Comments:
70	112	8 blank lines
70	112	8 blank lines
70	112	8 blank lines
47	71	L.M.S.: Load Memory Scan
20	32	tells ANTIC that screen
3C	60	memory starts at \$3C20.
37	55	
37	55	display ANTIC mode 7 (GR. 2)
37	55	bits 4 & 5 set, to
37	55	enable fine scrolling
37	55	
37	55	
37	55	
37	55	
37	55	
37	55	
41	65	JVB: Jump to Vertical Blank
00	00	and start of display list,
30	48	which is at \$3000

For most games, the background will be one of the special character set modes (ANTIC modes 4, 5, 6, 7) used with a custom character set. Examples of games using this technique are Eastern Front, Protector, Wizard of Wor, Pac-Man, Centipedes, Nautilus, Canyon Climber, Miner 2049er, etc. Almost every commercial game uses a custom character set background. However, to save time and space, our fine scrolling demo will just scroll over the data in the operating system. To make your own background, you must (1) define your character set, then (2) place your screen data into memory someplace, and (3) change the Load Memory Scan pointer in the display list to point to your screen data. Custom displays take a lot of time, but the results are well worth it. A good background is a must for any commercial game.

The final step is to write a routine to do the scrolling. There are two fine scrolling registers in the operating system called HSCROL and VSCROL. HSCROL controls horizontal scrolling, and VSCROL controls vertical scrolling. To use them, simply poke or store the number of lines you want scrolled.

There's just one problem: fine scrolling can only scroll so far. The vertical limit is 16 TV-scan lines, and the horizontal limit is 16 graphics 7 pixels. If we attempt to scroll beyond those limits, ANTIC simply ignores the higher bits and starts over at zero.

The solution is to modify the Load Memory Scan instruction address in the display list. The LMS instruction contains a 2-byte address which tells ANTIC where in memory to start fetching data for the screen. If we decrease the value of that 2-byte address by one, the whole screen does a coarse scroll to the right. If we increase it by one, it coarse scrolls to the left. Similarly, if we add 40 to it (using a graphics mode that requires 40 bytes per line), the screen scrolls up. And if we decrease the address by 40 (using the same graphics mode), the screen scrolls down.

Notice the word 'coarse.' Scrolling using the LMS pointer has no limits, but the screen moves in big jumps. To get unlimited fine scrolling, we must combine the fine scrolling registers with coarse scrolling using the LMS pointer.

So, to do fine scrolling, we first fine scroll the screen, keeping track of how far it has been scrolled. When the amount of scrolling equals the size of the pixel or character, we reset the fine scroll register to zero and change the LMS address to execute a coarse scroll.

Look over the scrolling program itself and give it some study. When scrolling to the right, the HSCROL register is increased each time until it reaches a value of 8 (because graphics 2 characters are 8 pixels wide). Then, HSCROL is reset to zero and a coarse scroll is executed by adding one to the value of the LMS address. The same technique is used for the other 3 directions, and diagonal scrolling is done by combining two directions into one scroll. For example, to scroll diagonally up and left, a 'left' scroll and an 'up' scroll are both executed during the same vertical blank.

Feel free to incorporate this scrolling program into your own game. In closing, here is short goodie you may want to type in and try:

```
100 COLOR EQ $F0
110 OR $600
120 LOOP LDX COLOR
130 INX
140 STX COLOR
150 STX $D018
160 STA $D40A
170 JMP LOOP
```

It's a good effect for the title of a game, although you'll have to rewrite it so you don't need to press system reset.

Until next month, good luck and great gaming!

—Jon Attack

# Ockers

## Machine Language Programming #3

Compare Instructions and Hooking to Basic

In the last session we created a loop using a branching instruction. Because of the way the loop was terminated, it was necessary to count backwards with the index. There should be some way of terminating a loop while counting forward and there is. The COMPARE instructions (CMP, CPX and CPY). Compare instructions involve one register (Accumulator, X or Y) and one other byte. Where the other byte comes from is determined by the addressing mode. In a loop using X as a counter we use CPX:

```
LDX # $00          ; Initialize count LOOP
INX                ; Next
CPX # $14          ; Finished?
BNE LOOP           ; Not yet
```

The instruction CPX # \$14 compares the X register with the hex number \$14. It DOES NOT affect the X register but it DOES set and reset flags. Remember the branch instructions operate depending on the condition of the flags. We have mentioned the zero (Z) and negative (N) flags. I want to introduce another, the carry (C) flag. We won't say what it is until we discuss binary arithmetic; it's sufficient to know this flag, like others, will be set or reset depending on the previous operation.

The branches to go along with the carry flag are:

```
BCC Branch on Carry Clear (C = 0)
BCS Branch on Carry Set (C = 1)
```

By using the flags after a compare operation we have the equivalent of Basic's IF ... THEN construction:

Want to Branch if:	Then use:
REGISTER = DATA BYTE	BEQ
REGISTER <=> DATA BYTE	BNE
REGISTER < DATA BYTE	BCC
REGISTER >= DATA BYTE	BCS
REGISTER > DATA BYTE	BEQ
	BCS

(BEQ branches over BCS).

Let's create a program to handle joystick movement of a cursor. Suppose we want only orthogonal movement (no diagonals). One solution is given in listing 1. The routine itself is quite simple, it compares the joystick value it gets to those it expects for certain directions and updates the direction only if they correspond. Notice the PLA at the beginning and the RTS at the end. These are used in 'hooking' the machine language routine to Basic. If you just put the routine itself in machine language with a loop from the end to the beginning, it will run too fast. With the USR function in Basic we can call the routine only when we want to, one time through for each call. Just put the routine in page 6 using 'Hexpoke', then type NEW and run the following Basic Program:

```
100 TRAP 100:POKE 752,1
110 PRINT " ";CHR$(30);
115 FOR J = 1 TO 10:NEXT J
120 PRINT " ";CHR$(30);A = USR(1536)
130 FOR J = 1 TO 10:NEXT J
140 GOTO 110
```

The CHR\$(30)'s (backspace) are there because the cursor column automatically advances once after the print. Leave them out and you won't be able to move left. The trap is to prevent a 'cursor out of range' error message. Play with the delay loops to see their effect on the timing.

If you want to use this routine in a Basic program you won't want to load 'Hexpoke' and punch it in every time. We need the routine itself in the Basic program. One way to do this is to POKE the data in as follows. The Hex numbers are converted to decimal, (I use the program on page H-18 of the Basic Manual), and put into Data statements.

```
10 RESTORE 20:FOR J = 0 TO 28:READ A:POKE 1536 + J,A:NEXT J
20 DATA 104, 173, 120, 2, 201, 14, 208, 2, 198, 84, 201, 13, 208, 2, 230, 84, 201, 11,
208, 2, 198, 85, 201, 7, 208, 2, 230, 85, 96
```

It is nice to keep page 6 free for other uses. In this case I tend to put my machine language routines in strings as follows:

```
10 DIM STK$(29):RESTORE 20:FOR J = 1 TO 29:READ A:STK$(J,J)
= CHR$(A):NEXT J
```

The function is then called with A = USR(ADR(\$\$)). This only works if the routine is completely relocatable and is able to reside any place in memory. This means you can't use any instructions which depend on where the program is located. Absolute jumps to the program itself are out. These change depending on the program location. Relative branches are O.K. as are references to locations which don't change (STICK0 for example).

### Problem #5

We used a TRAP in Basic to handle any out of bounds cursor movement in our cursor program. A better way is to handle the problem at machine level. Re-write the program of listing 1 so that top, bottom, left and right limits to cursor motion are included.

	COMPARE Instruction OP codes		
addressing mode	IMMED.	ABSO.	ZPAGE
CMP	C9	CD	C5
CPX	E0	EC	E4
CPY	C0		

—Stan Ockers



## BRIAN'S ARCADE

Along with the New Year there have been some pretty good games to go along with it! **The Sands Of Egypt** is a new adventure game from Datasoft Inc. I can't give the game a full review because I have not gotten very far in the game. I am going to give the game to Jim Bumpas, our co-editor, and he will give it a full review next month!

One of the things I really like about the game is the superb scrolling. When you move west, for example, the entire picture scrolls to the left. So far I have only three different scenarios and I've heard no sound. But I don't know, because like I said I haven't gotten very far. If you want to play a tough adventure game you should buy **The Sands of Egypt**.

Another game we recieved to review is **Serpentine** from Broderbund Software. Serpentine is another maze type game. Hold on, I know what you're thinking: Another Pac-Man rip-off, right? Wrong!!

Serpentine is a totally original game. In the game you are a blue serpentine. Your enemies are the orange serpentes. You travel around the maze while trying to kill the orange serpentes. The only way to kill them is to eat them from behind until they turn green. When the orange serpentes turn green then you can eat them head on or from behind. You get more points from eating them head on, though, and you also make your serpentine one segment longer.

The graphics in serpentine are not awesome, although they are pretty good. I really like the sound in serpentine. The sound I like the best is the sound the serpentes make. Serpentine is available from Broderbund Software on disk for about \$30. If you are tired of the same old maze game you should definitely buy Serpentine, you'll love it.

Adventure International was kind enough to send us a review copy of their new game **Sea Dragon**. So here it is. If you like Scramble or Super Cobra in the arcades then you will love Sea Dragon.

In Sea Dragon you control a nuclear sub named Sea Dragon. Your task is to travel deep under the ocean avoiding treacherous mines, depth charges, laser bases and ships which drop bombs on you. You must eventually get to the Master Mine and destroy it!

The graphics aren't great but they're pretty good. One thing I like is the color animation and the sound. You can actually hear the sonar as you go about destroying mines, etc. If you like the arcade version of Scramble or Super Cobra or the home version of Air Strike then you will really like Sea Dragon.

That's all for this month. Next month I'll try to review Miner 2049'er, Way Out, Defender and anything else I should receive.

See ya later,

—Brian Dunn



## New Atari

The biggest news is the release of the **Atari 1200XL**, a sleek new 64K 6500 computer with 12 user programmable functions, built in diagnostics, and compatibility with all Atari Hardware and Software. It has only one cartridge slot, 2 joystick ports, 256 colors, 3½ octave sound range, and some new graphic modes. The extra memory, of course, doesn't make much difference, since the Operating System, etc., still takes up the same amount. Did you know that the VIC 64 has only 38K available to BASIC even without DOS? Price "under \$1000"—probably \$800.

Also released are the new printer, the 1025 for \$549, which is an Okidata; a 40-col color printer, the 1020, which for \$299 gives 4-color printouts on 4½" paper; and a new cassette recorder, the 1010 for \$100.

## Be A Writer!

Becoming a published author in software is not as difficult as it might seem on the face of it. Let me now outline some of the fundamental requirements.

Get the magazine Style Sheet. All magazines have them. The editor will send one with an SASE. **COMPUTE!** and **SOFTSIDE** have both more or less published theirs although I do not have the issue numbers in front of me.

Fundamentally, the requirements are for wide margins and double spacing and no right justification. This supports their estimate of how many column inches your article will fill. Then each page must have your name, a short title, and a page number. All programs, diagrams, and screen dumps should be clearly labeled. There is a very simple reason for this. Manuscripts get dropped and mixed up with others. This is for your benefit as much as theirs. There are other requirements but the point is most any word processing program can handle these basic requirements.

Most of my experience has been with **COMPUTE!** This is due to the delay time in getting a response from the magazines. I have moved on to **COMPUTE!** rejects but have yet to receive a response. But from general knowledge of magazines let me say a few words about what I have learned.

The average magazine buys from three to six issues prior to publication. Also they usually take from two to four months to respond to a manuscript submission. This means the manuscript submitted in January and bought in March may not see print until September. There is something to be said for having a long pipe line. But what does "buy" mean. Unless you work a special deal it means that you get so many dollars per page. If it is good enough to print it is good enough to get the standard rate but rarely is there any special consideration for an especially good article. If there is a reward it is in Royalties on the republication in "Best of..." books. But when do you get paid? Guess again. There are three common payment forms: upon acceptance, binder with remaining amount upon publication, and payment upon publication. Ordinarily the more established a magazine the more likely they are to pay upon acceptance. The new magazines are likely to pay upon publication. Do not expect payment will be in cash dollars US. In software magazines these can be in merchandise certificates which cost the magazine only the discount prices which they can arrange through their advertisers, although you redeem against the list prices. This, of course, is again limited to those just getting off of the ground.

A new writer needs a reputation. If one magazine will not buy then move on to the next and the next. Do not rewrite one word unless the magazine obviously is a different market. A simple graphics program rejected by **COMPUTE!** might be rewritten for **COMPUKIDS**. A new writer also needs experience and needs to interpret experience properly. The stories are legend about best sellers rejected by all but the last publisher. They are true. This leads me to the next aspect of publishing.

As with the publishing house, a magazine must have a broad audience if it is to stay in business. There is little concern for Best Seller potential in a publishing house and there is little concern for the neatest software trick this side of Sunnyvale if they have already bought up a year's supply of articles on the subject. I recently had a program named **SIMON** (in ACE Library) rejected twice in a row. **SOFTSIDE** ran a similar article in August and I have yet to see what **COMPUTE!** will run or maybe they just have enough games for the year. The point being is a magazine must have articles for the beginner, the experienced, and the seasoned user and must have articles for young and old. It must have utilities, games, tutorials, business and misc. To write you can write on any subject for any age group and to sell the most I suggest you write for all categories.

What is most important? A professional editor and I are certain the amateur editors will tell you the primary problem is filling the magazine or newsletter. First it is hard to get people to write and second what is written must be written in coherent English with reasonably proper grammatic construction and spelling. This is every editor's problem. Next there is interest in the subject matter as publication material. A computer magazine editor has one further problem, the software must run with minimum problems. It pays to thoroughly debug anything you submit. No editor wants to publish a list of corrections to your program as long as the original article although it has been done.

A manuscript will consist of two parts—the prose and the program (the latter may not be needed). The prose is the documentation. It tells the user of the program what it will do, why, and how to use it. It covers all aspects of what it can do and how to get the best use of it. Also it provides sufficient information to change it to fit individual needs if that is appropriate. With software nothing, absolutely nothing, is obvious. The Navy uses MIL-STD-1679 to guide not only the documentation of software but the excruciating detail as to how it must be written. This is not for fun. This is so someone besides the author can make some sense out of it. For example, get out a program you wrote a year ago and haven't looked at since. Explain it. You will find you can not understand what you did without work. Imagine your reader who never even saw it work before. You are really selling the documentation for your program. The magazine will determine whether or not the subject is of interest to them.





# CAMPFIRE MIXERS

The computer itself is only 36 years old. Affordable and useful computers are only 20 years old. The personal computer is 8 years old. The Personal Computer software publishing business is 6 years old. ATARI personal computers are three years and the first seriously published article containing software for the ATARI was in late 1980 (I think.) It is often difficult to put into perspective just how much has occurred in such a short period of time. Think about something else you did three or eight years ago like a marriage or a child being born and it seems like only yesterday. Yet my son was born about one year after I decided against buying the first ALTAIR computer. SOFTSIDE magazine has published 31 issues, A\*N\*A\*L'O\*G about six, ANTIC two, (COMPUTE I can't figure out perhaps as few as 31, as many as 43.)

If you consider publication of software articles then keep in mind you are not submitting articles to The New Yorker or the Atlantic Monthly nor are you dealing with OMNI, National Lampoon, or Heavy Metal —not that I wish to discourage you. The software publication business is barely beginning to be defined as a market. You can still afford to subscribe to every computer magazine published, almost. Those covering the ATARI you can almost get by dropping hints to friends and family around Christmas and your birthday.

Next let us look at the publishers addresses. COMPUTE is in Greensboro, North Carolina (probably just down the road from the Mother Earth News). SOFTSIDE is in Milford, New Hampshire. These addresses are a far cry from Lexington Ave, in New York City. What has happened? Well the first thing to think about is: Eugene, Oregon is no longer the least likely place to start the hottest software magazine. This is a new era of publishing perhaps more profound than the development of science fiction and perhaps even more important.

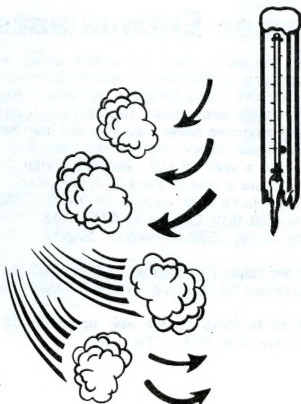
In 1938 (or 1936) John W. Campbell became the editor of Astounding Stories and single-handedly turned around the field. He changed what was a scientific romance parallel to a gothic romance into a field which required true solid understanding of science along with good fiction.

What is happening today? A magazine editor may no longer simply know his skill as writer and as an editor, he must be skillful in a way which the science fiction editor never had to be. John Campbell was fond of requiring his writers to be as solid in their science as they were in their fiction. Today's software magazine editors have an added challenge. Even John Campbell let more than a little poor science slip by him and in some areas he encouraged it. Today's software magazine editors can let nothing slip by them. Why? Because the computer is available and the programs are submitted on magnetic media. There may have been pseudo-science but there are no pseudo-programs.

Today we have a discipline introduced to publication like never before, a solid discipline where the computer is the final judge. The writer and editor face a dual challenge. The skill in programming or knowledge of the system is as important as the literary skill in putting together a manuscript. This is a new field, a virgin field which is open to anyone who has the skill to program. Today's old hands at science fiction started as teen-agers and they knew how to be creative within the restrictive framework of science. They had barely grasped the foundations of writing. They were accepted because the science was more important than the fiction. Today the program is more important than the article which goes with it but both are important.

What we have developing is a form of disciplined writing for the average person which was previously unseen outside of the professional literature of the sciences. Its expansion will probably directly follow the expansion of personal computers. This is a wide open field which will become more and more professional but will always have room for the new comer. Anyone who masters a solid understanding of the computer and of software, and who can meld this skill with writing skill has a wide open potential to operate in this new, and fast growing market. There is a future here and it is open to anyone willing to take advantage of it.

—Matt Giwer



Since school began in September, we have grouped a combination of 6 boys and 6 girls ranging in 4th through 6th grades for a Campfire Club. We have titled ourselves "The Campfire Mixers"; and for our Indian name: "To ka ta pa", meaning, "In the future".

Heading our club is Jackie Heden, with special interest and helps, her husband, Richard, and myself, Nancy St. Onge as Assistant leader.

Since we have a special interest in computers Jackie and I decided it would be fun as well as educational to choose computers as the theme of our club. Our group became very excited at the thought of this, and are very eager to learn this "hands on" experience.

While our computer plans are in progress, our club went on their first excursion to Camp Wilani for an overnight campout in veneta.

Kirt Stockwell, president of ACE came as a special guest to one of our meetings. He demonstrated many different types of program languages, the uses of a Modem and how to use a Word Processor. Ruth Ellsworth demonstrated the Pilot Language.

On one of our field trips to Lane Community College, Kirt presented a surprise to our Campfire Club. As we came into the forum and sat down, the children began to read, moving across the screen; "ACE WELCOMES THE CAMPFIRE MIXERS, WHO ARE IN THE FUTURE —" followed by each one of the children's names, one by one flashing in different colors across the screen. The children shouted and giggled as they watched and felt very honored. Kirt also programmed our symbolgram for our club which consisted of four different Indian symbols. They are person, hand, thought and eye which are set in a square representing a computer. Located at the bottom is our Indian name: "to ka ta pa".

Our children are slowly beginning to understand within their own young minds the reality of what a computer can do. One of their favorite programs is called "The Talking Head" which is demonstrated using sound. Some of the other field trips we have taken the children on were to the University of Oregon Computer Center where Floyd Bard demonstrated the different uses of various computers. One of his special surprises to the children was a Computerized Snoopy which he presented to each child personally. We also went to Video Toones and Marty let our children play several video arcade games.

In December we took the children to The Willamette Institute of Science and Technology where they had many attractive displays of different computers, games, and other interesting educational things for various ages of children. Many children entered Wistec's Video tournament in hopes of winning an Atari computer and other prizes which were given away. Wistec's Planetarium Show of the Stars and Planets highlighted our trip there.

At our last ERACE meeting we talked to Jim Schwartz who is a High School Teacher at Cal Young High School. Jim is currently experimenting with basic programming with his children in the classroom. He shared many ideas with us to teach our children a fun and simple way to learn Basic Programming. We really appreciate Jim's help because it is very difficult to find ideal programs to teach children how to understand the basics which are needed in learning to operate a computer.

Towards the middle of December our club had a Bowl-A-Thon to raise funds which were needed in our treasury. We are happy to report this was a great success. We raised \$400.00 which was a great boost to all of us. Some of the things we want to accomplish include purchasing an individual disc for each child upon which to learn programming; patches and group beads which are earned by the group and two special field trips which the children choose for our club. Our foremost goal is an overnight excursion to the Coast in April. This is designed to reward our boys and girls for their cooperation and patience which was needed in order to teach them the basics of computers.

In January, Campfires will be having their annual Candy Sale and our club will be looking forward to supporting this good cause to which we belong.

Also our club has been invited by Don Marr, with Royal Software to come and observe how computers are used in his business.

As February comes near our club will be very busy preparing for the International Faire which will be on March 5. Each child will select the country of their choice, design an outline of their country and flag, and will select a unique feature which impresses them the most about their country. Then Richard will help them program this information and the children will display and demonstrate their computer programs at the faire.

Also our club will be having our Ceremonial in February. This is to reward our children for their past efforts and accomplishments by presenting them with Try-Ad Beads. The children are learning to program and by accomplishing certain tasks on the computer successfully will earn different colored beads that represent what they learned.

On May 7, we will be having our last and final Grand Council Ceremonial for the entire year, where all the clubs of Campfires will attend. There will be approximately 2,000 children there. Also we will have a lighting ceremony and many prominent advisors and leaders will be attending. At this time, we want to extend our invitation to ACE to be a part of our wonderful Computer Try-Adventure program by attending this Grand Council Ceremonial.

Through the excitement and happenings unfolding before us, we have really felt a surge of hunger for the knowledge to learn the computer, and the urge to push forward; to continue and keep learning this Keyboard Wonder!

When we started our unique little club we went in blindly, not knowing how to supervise or direct this type of club; but through the kindness, help from others, time and prayers which were contributed we have been very successful so far in the past four months.

Our gratitude is extended to the following people in sponsoring us in the Bowl-A-Thon; and helping us to reach our goals: Jack Ball, Manager of Southtowne Lanes, Members of ACE, ERACE, Video Toones, People of Pacific Northwest Bell on Grant Street, Graham Kerr, formerly the chef on "The Galloping Gourmet," the childrens teachers, friends, relatives, Courtney Arnold, our Supervisor in Campfires, and any others who deserve our sincere thanks.

—Jackie Heden, Nancy St. Onge



MEETING: FEB 9, 7:30  
L.C.C. FORUM 308  
(south center parking lot)

## Bulletin Board (503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400, 2 double-density disk drives, an Atari 825 printer, a Hayes SmartModem; running the ARMUDIC Bulletin Board software written by Frank Huband, 11066 N. Stafford St. Arlington, VA 22201

## TYPESET COPY From Your Computer

If you have a modem, word processing and communications programs, you can transmit your manuscript, article, newsletter, or other text via telephone and have it converted into space-saving, easy-to-read type! Computer typesetting offers you high quality copy at "do-it-yourself" rates: just \$2.00 per 1,000 characters. And it's easy to encode your copy with our designated typesetting commands. Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." Write or call Dennis Hunt for a brochure and simple instructions.

**EDITING & DESIGN SERVICES**  
30 East 13th Avenue Eugene, Oregon 97401  
Phone (503) 683-2657

ATARI  
COMPUTER  
ENTHUSIASTS  
3662 Vine Maple Dr. Eugene OR 97405

# FIRST CLASS MAIL

## Atari Computer Enthusiasts

A.C.E. is an independent computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

**All our articles, reviews and programs come from you, our members.**

Our membership is world-wide in scope; membership fees include the A.C.E. Newsletter. Dues are \$10 a year for U.S., and \$20 a year Overseas. Airmail and include about 10 issues a year of the ACE Newsletter. President—Kirt Stockwell, 1810 Harris #139, Eugene, Or 97403 503-686-2470 Secretary—Charles Andrews, POB 1613, Eugene, Or 974401613 503-747-9892 Librarian—Chuck and Jody Ross, 2222 Ironwood, Eugene, OR 97401 503-343-5545

Editors—Mike Dunn, 3662 Vine Maple Dr., Eugene, Or 97405 503-344-6193 Jim Bumpas, 4405 Dillard Rd., Eugene, Or 97405 503-484-9925

Send a business-size SASE to the Ross' for the new, updated ACE Library List!!! Best of ACE-1981, still available, Disk or Tape, \$8!